

Chatbot General Review

What did we learn last week?



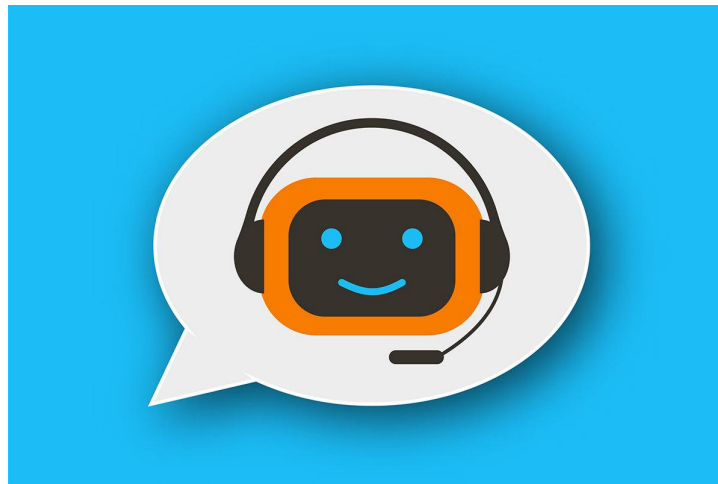
Attendance



What is a Chatbot?



“A computer program designed to simulate conversation with human users, especially over the internet”



Categorizing Tasks

We'll categorize our chatbot's tasks into three categories: tag, patterns, and responses

- ▷ Tag: General summary for the task
- ▷ Patterns: User sentence inputs that relate to these tasks
- ▷ Responses: How the bot should respond to this input

Example File (Intents)

Below is some example data for our weather bot tasks:

```
{
  "intents": [{
    "tag": "clothes",
    "patterns": ["What should I wear today?", "What clothing would be appropriate?"],
    "responses": ["You should wear shorts and a t-shirt", "Light clothes are what you need"]
  },
  {
    "tag": "weather",
    "patterns": ["What's the weather today?", "What's it like outside?", "Tell me about the weather"],
    "responses": ["It's sunny today!", "The sun is shining!", "It's a warm and clear day"]
  }
  ]
}
```

The response values don't have to be hard-coded. In this example, it would make more sense to get weather data from an external source, like an API

Bag of Words Approach

A simple but effective approach we can use is the bag of words model

The main idea is to train a model with the **patterns** as the inputs and the **tag** as the output



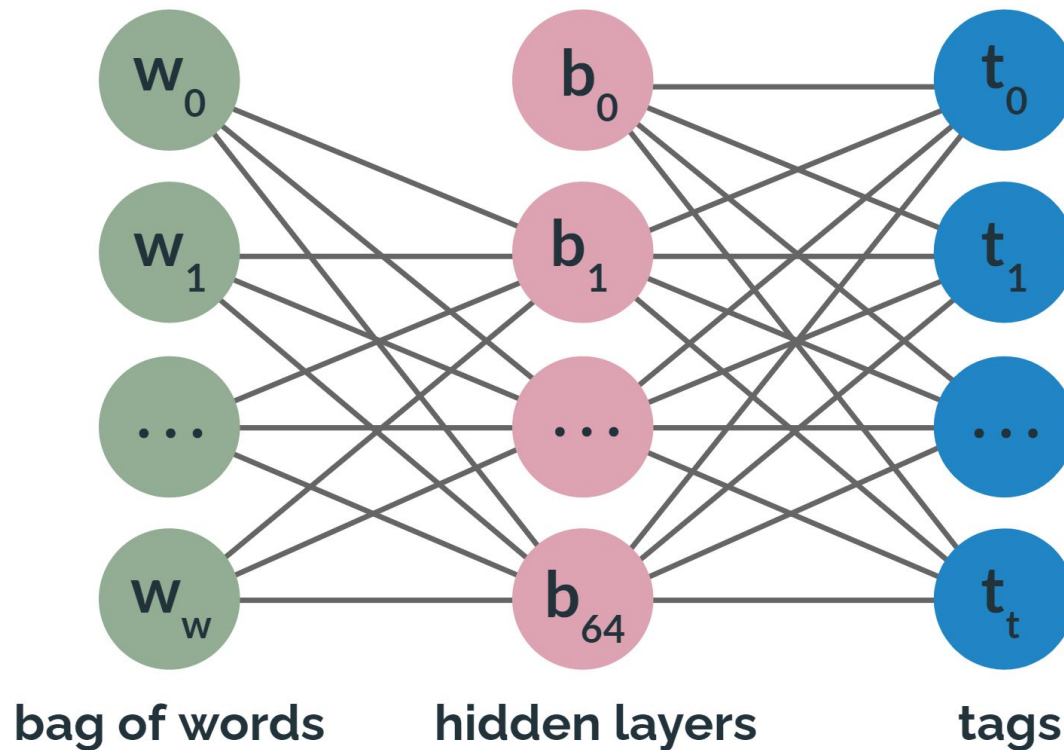
Bag of Words Algorithm

In general, to create a bag of words we need to:

1. Tokenize and clean every word in the patterns
2. Add these words to a list that we'll call our bag
3. For the user input, create a list of 0s and 1s, where 1 means the word is in our bag, and 0 otherwise
4. Create an output vector corresponding to our tags

Neural Network Training

Now that we have numeric inputs and outputs, we can train our model using a neural network!



(Don't worry I've coded all of this for you in the notebook)

Questions About Anything?

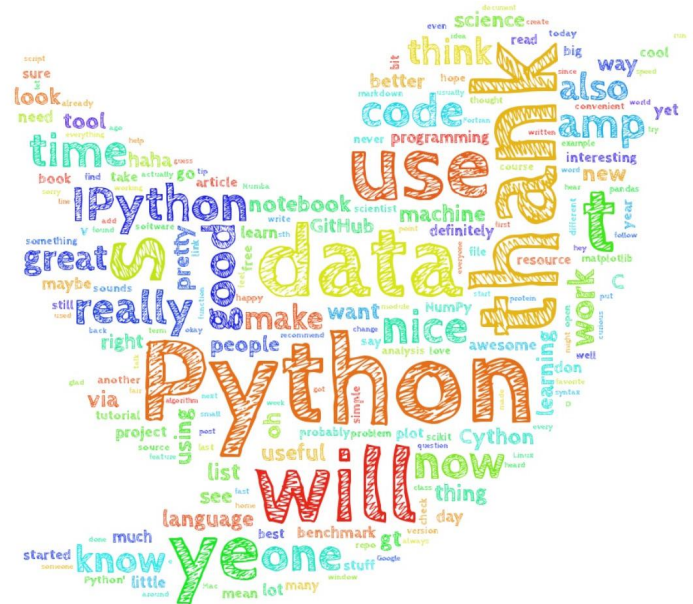
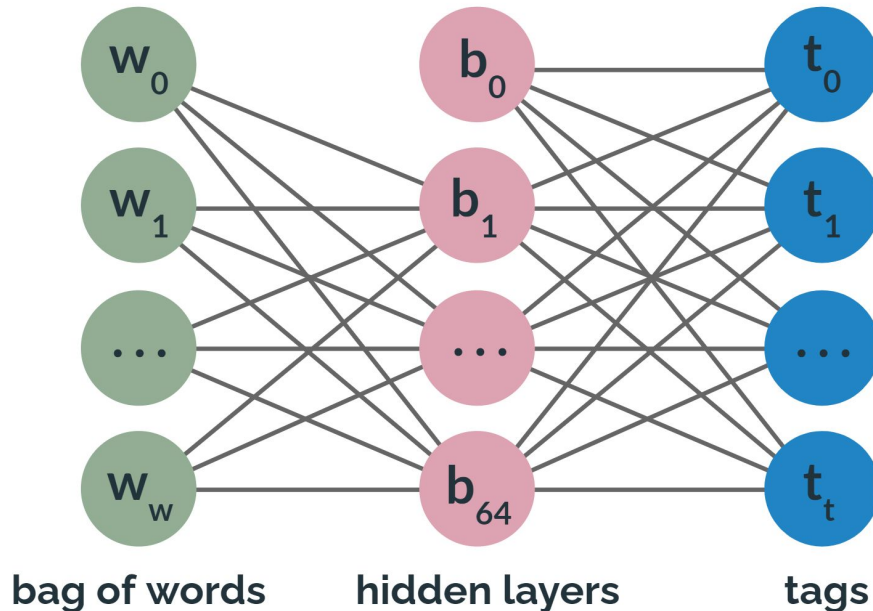


Making Predictions

How do we make predictions using bag of words?

Bag of Words Predictions

We'll now use the bag of words model combined with our neural network to make predictions!



Prediction Algorithm

In general, to make a prediction from a user input:

1. Create a bag of words from the user input
2. Forward propagate with our neural network
3. Analyze the tag outputs and choose the best one
4. Do something with this tag (up to programmer)

Example: Bag of Words and NN

Let's consider the user input:

```
"Hello, how are you"
```

Convert to a bag of words:

```
bag = [1, 0, 0, 0, 1, ..., 0]
```

Predict with our neural network:

```
numpy: forward_prop(bag)
```

```
keras/tensorflow: model.predict(bag)
```

Numeric Outputs (Review)

Suppose our tags are:

`["greeting", "goodbye", "clothes", "rain"]`

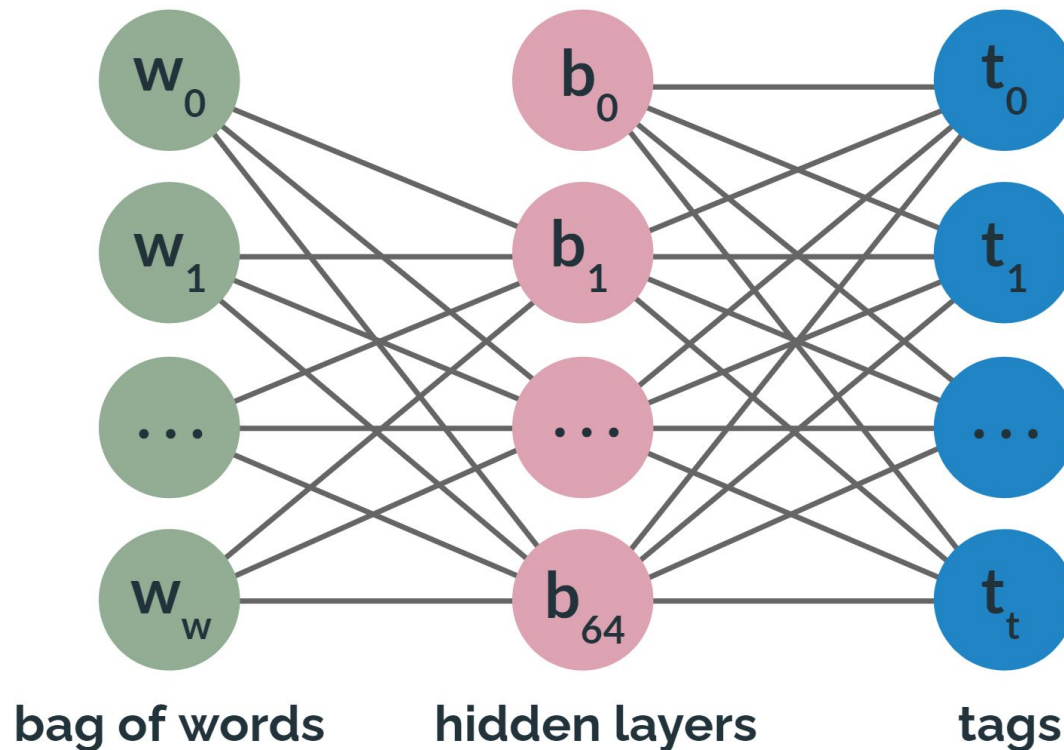
Which tag would these vectors correspond to?

`[0, 1, 0, 0]`

`[0, 0, 0, 1]`

Multi-Classification

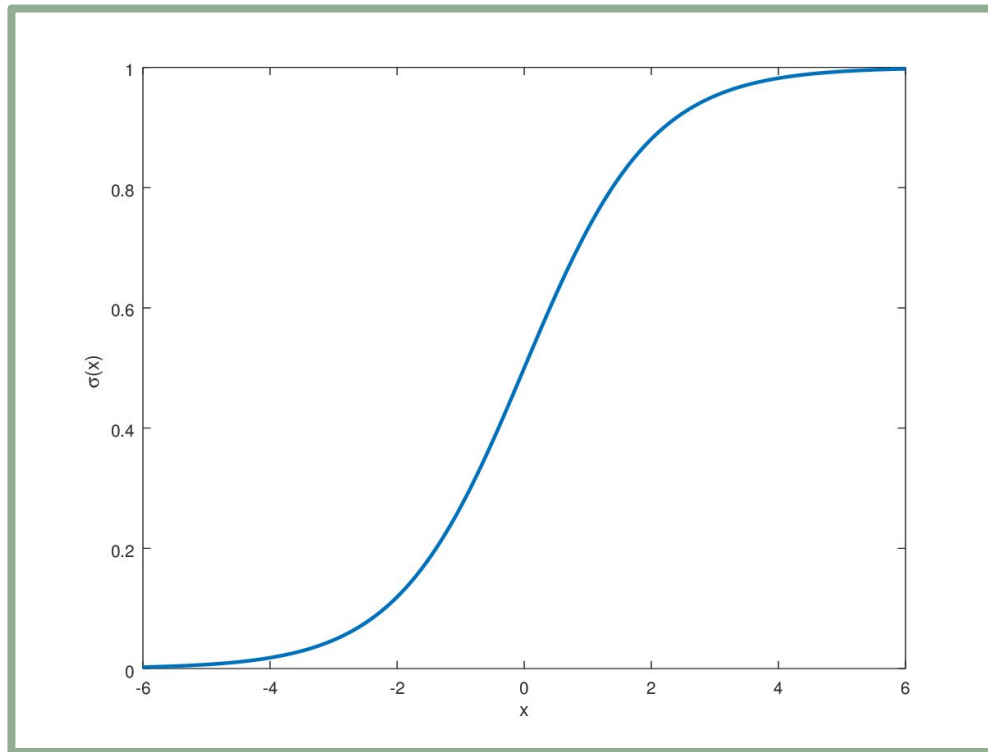
Recall our neural network architecture:



What do we know about the values in tags? (Hint: think sigmoid)

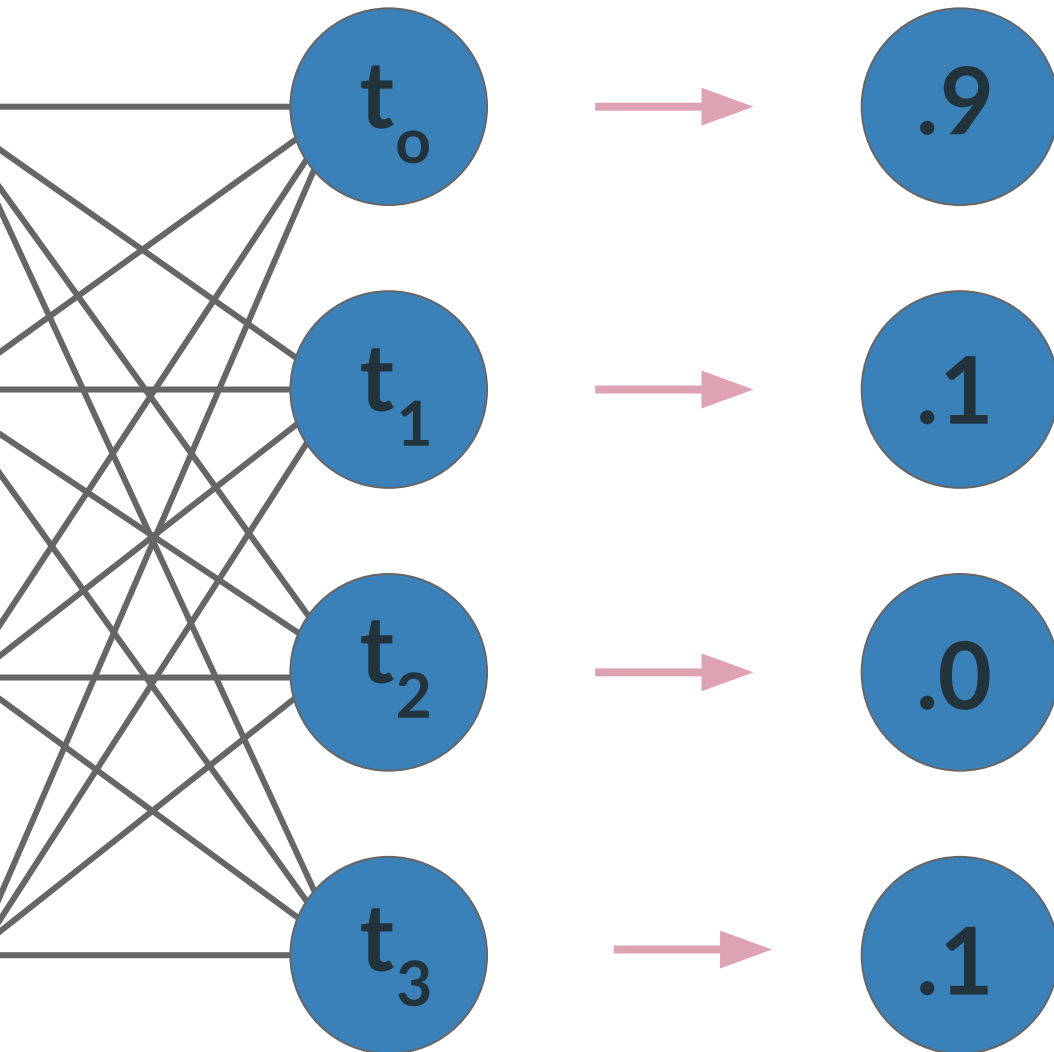
Multi-Classification

Here's the shape of our sigmoid function:



Because we apply this function, our tags will always be between 0 and 1!

Example: Analyzing the Output Layer



Which tag
should we pick?

Example: Analyzing the Output Layer

From last slide, we calculated our tag values as:

[0.9, 0.1, 0.0, 0.1]

We determined that the tag with value 0.9 would be the best pick because the bot was the most confident

Thus, if our tags are:

["greeting", "goodbye", "clothes", "rain"]

We predict the tag to be greeting

Review: Parsing intents.json

Last, week, we created a response dictionary with the following schema:

Tag	Patterns	Responses
greeting	["Hello", "Hi", "Howdy"]	["Hey!", "Hi, human scum"]
goodbye	["Bye", "See ya", "Goodbye"]	["Adios amigo", "Until next time"]
clothes	["What should I wear?", "Give me fashion advice"]	["Go for a t-shirt and shorts"]
rain	["Will it rain?", "How are the clouds looking?"]	["Yes it will rain", "It's mad wet"]

Example: Retrieving a Response

Now, with our predicted tag, we can get a response from the dictionary!

```
predicted_tag = "greeting"
```

```
response_dictionary[predicted_tag]
```

```
> "Hey!"
```

Additional Note: More Complex Responses

If you want, you can have more complex responses with additional logic

```
user_input = "Will it rain?"  
# gets the weather from an API  
if (predicted_tag == "rain") {  
    api.get("weather.com/uiuic")  
}
```

Tasks to Complete



- 1) Work on the notebook (chatbot_pt_2.zip) in the google drive folder

<https://drive.google.com/drive/folders/1Qr4AaFRgTn8kYkQxHkz2edwGO26MQm4Z?usp=sharing>

Note: There's two notebook implementations, one that uses Keras/Tensorflow and one that doesn't. If you want to experiment with machine learning libraries, use the former. If you don't want to install any additional packages, use the latter

Try to work on it collaboratively! You might meet some people you could do a project with in the future

As always, let us know if you need any help!

SIG

NLL

