Deep Learning and NLP A-Z

Q & A

All the answers to your questions, section by section

SuperDataScience Team

Abstract

First, we would like to thank all of the students who have contributed to the discussions and Q&A sections. It's always a great idea and method of enhancing the materials to be able to discuss key concepts. In the following PDF you will see some of the main questions that we see frequently asked. If you don't see the answer that you are looking for or if the section doesn't have any information for the FAQ please post in the Q&A in the course. In addition, the following section has some of the main setup information and general questions for the entire course:

- **I'm having some audio and video issues, is there any recommended troubleshooting steps?** Yes, you can use the following steps provided by Udemy at:

  `https://support.udemy.com/hc/en-us/articles/`
  `229231227-Video-Audio-Issues-Troubleshooting`

  and at

  `https://support.udemy.com/hc/en-us/articles/`
  `360003054894-Troubleshooting-Issues-With-Udemy-s-App`

- **What IDE is the instructor using?** We are using Spyder inside Anaconda. You will set it up in Section 3 lecture 17: ChatBot – Step 2.

- **How can I activate my environment that I installed?** For Mac and Linux you can use the command: source activate my_environment_name and for Windows: activate my_environment_name

- **What version of Python and Tensorflow are we using to install the environment?** We will be setting up Python 3.5 and Tensorflow we use 1.0.0

Please see the Table of Contents section below for a breakdown on the contents of this FAQ document.

Table of Contents

Summary

1 Section 1 - Welcome To The Course

1.1 Lecture 1 - Get Excited!

1.2 Lecture 2 - Applications

1.3 Lecture 3 - Some Additional Resources!!

- **Where can I obtain the additional resources for the course?** Resources can be obtained at the following link: `https://www.superdatascience.com/deep-learning-chatbot/`

2 Section 2 - Deep NLP Intuition

2.1 Lecture 4 - What You'll Need For This Module

2.2 Lecture 5 Plan of Attack

- **Where are the annexes located?** The Artificial Neural Networks Annex 1 is located in Section 10, and you can find the first lecture for this Annex at: `https://www.udemy.com/chatbot/learn/v4/t/lecture/8808280?start=15`

- Annex 2 Recurrent Neural Networks is located in Section 11 starting with: `https://www.udemy.com/chatbot/learn/v4/t/lecture/8805309?start=0`

2.3 Lecture 6 - Types of Natural Language Processing

2.4 Lecture 7 - Classical vs Deep Learning Models

2.5 Lecture 8 - End-to-end Deep Learning Models

- **Is there a bonus for using end-to-end models?**Using end-to-end models can help eliminate the need for more traditional task related and has seen a high boost in performance and benchmarks.

2.6 Lecture 9 - Bag-of-words model

- **What is the difference between a bag of world model and vector space model?** For the comparison between the models we can see it mentioned in the following discussion that notes:

  Bag-of-words: For a given document, you extract only the unigram words (aka terms) to create an unordered list of words. No POS tag, no syntax, no semantics, no position, no bigrams, no trigrams. Only the unigram words themselves, making for a bunch of words to represent the document. Thus: Bag-of-words.

  The Vector space model: Given the bag of words that you extracted from the document, you create a feature vector for the document, where each feature is a word (term) and the feature's value is a term weight."

2.7 Lecture 10 - Seq2Seq Architecture (Part 1)

2.8 Lecture 11 - Seq2Seq Architecture (Part 2)

2.9 Lecture 12 - Seq2Seq Training

- **Is there any additional resource or information for the process of seq2seq training?** Yes, the following resources can provide further context:

  `https://www.tensorflow.org/tutorials/seq2seq`

  `https://google.github.io/seq2seq/`

2.10 Lecture 13 - Beam Search Decoding

- **Do we use beam search decoding throughout the algorithms in the course?** We do use beam search in a recently added version of the Chatbot that was built by Abraham Sanders. It even has the option to change the beam search values on the command line to visualize pre-trained weights with the chat output that makes it a great resource to have to examine beam search further.

2.11 Lecture 14 - Attention Mechanisms (Part 1)

2.12 Lecture 15 - Attention Mechanisms (Part 2)

3 Section 3: Building a ChatBot with Deep NLP

3.1 Lecture 16 - ChatBot - Step 1

3.2 Lecture 17 - ChatBot - Step 2

- **What version of Python and Tensorflow are we using to install the environment?** We will be setting up Python 3.5 and Tensorflow we use 1.0.0

- **How can we create a virtual environment?** We first open the terminal or Anaconda prompt and launch the terminal/prompt. Now please run the command:

  conda create -n chatbot python=3.5 anaconda

- **How can I activate my environment in the terminal or prompt?**

  For Mac/Ubuntu you can run: source activate chatbot

  For Windows: activate chatbot

- **Where can I download the code for the course?** There isn't a link to download the code because you can get each part (for the sections) in the Checkpoint Lectures and can obtain the final code in the checkpoint at: `https://www.udemy.com/chatbot/learn/v4/t/lecture/8887796?start=0`

3.3 Lecture 18 - ChatBot - Step 3

4 Section 4 - Part 1 Data Preprocessing

4.1 Lecture 19 - Welcome to Part 1 - Data Preprocessing

4.2 Lecture 20 - ChatBot - Step 4

4.3 Lecture 21 - ChatBot - Step 5

4.4 Lecture 22 - ChatBot - Step 6

- **How can I run the code in Spyder?** To run the lines you can select them and then press ctrl + enter to execute the code.

- **What does adding encoding = 'utf-8', errors='ignore') do?** Simply, it's added so that any files (data,text,csv, etc) that are added or run for the program or for specific files are encoded properly (basically setting it to a language that can be understood by the program) since we have a range of file formats. But for a really in depth answers take a look at the following: `https://stackoverflow.com/questions/2241348/what-is-unicode-utf-8-utf-16`

- **Where are the movie_conversations and the movie_lines files located?** You can obtain them from visiting: `http://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html`

4.5 Lecture 23 - ChatBot - Step 7

4.6 Lecture 24 - ChatBot - Step 8

- **I'm receiving a keyerror, what can I do to resolve it?** The keyerror tends to be caused from a syntax related issue. The best option is to examine the checkpoint and compare it to your code to make sure that the syntax and code is matching.

4.7 Lecture 25 - ChatBot - Step 9

4.8 Lecture 26 - ChatBot - Step 10

- **I ran into the following error: NameError: name 're' is not defined, how can I resolve it?** As a first step please try re-starting Anaconda/Spyder. If it persists please start a Q&A thread for assistance resolving it.

4.9 Lecture 27 - ChatBot - Step 11

- **There is an overlap in the conversation when it goes back and forth for several lines, howcome?**Thats because we are shifting the sequences (questions or answers) by 1, therefore the previous answer becomes the next question after the shift.

4.10 Lecture 28 - ChatBot - Step 12

- **Are the questionwords2int and answerwords2int dictionaries identical?** Yes, when they were created they ended up being the same, because the previous answers of the previous questions become the next questions of the next answers. It is normal to have the same two dictionaries at the beginning. It is also advantageous and recommended to create two separate dictionaries so that you can apply different thresholds in order to filter out non-frequent words in the dictionaries.

4.11 Lecture 29 - ChatBot - Step 13

- **What does the <PAD>token do?** PAD is the padding that replaces the empty cells by using the token <PAD>

4.12 Lecture 30 - ChatBot - Step 14

4.13 Lecture 31 - ChatBot - Step 15

- **Why don't we attach <EOS>and <SOS>tokens for the questions?** We don't attach them because we only need them for the answers since the first element required by the decoder is the <SOS>token and the last one the <EOS>token.

4.14 Lecture 32 - ChatBot - Step 16

- **How can sorting by length of the question help?** It's attributed to reduce loss through reducing the use of padding to help speed up training.

4.15 Lecture 33 - ChatBot - Step 17

- **Are we sorting by the length of the question?** Correct, we are sorting by the length of the questions of each conversation and not by the length of the question words.

4.16 Lecture 34 - Checkpoint!

5 Section 5: Part 2 – Building The Seq2Seq Model

5.1 Lecture 35 - What You'll Need For This Module

5.2 Lecture 36 - Welcome to Part 2 - Building the Seq2Seq Model

5.3 Lecture 37 - ChatBot - Step 18

- **Why is the input 2 dimensional?** The input is 2 dimensional because the neural networks can only accept inputs that are in a batch, as opposed to single inputs. We must add 1 dimension corresponding to the batch.

5.4 Lecture 38 - ChatBot - Step 19

- **Why do we delete the last column of the answer before adding <SOS>in the process_targets function?** We have to delete the last column to preserve the max sequence length since after that we make a concatenation to add the <SOS>token at the beginning of the sequence, we must remove the last token before so that the sequence length doesn't go over the max sequence length.

5.5 Lecture 39 - ChatBot - Step 20

- **What is LSTM dropout?** Dropout is a technique used for regularization in neural nets and it can help in learning and preventing overfitting with the data. If you would like some further context please see the following resource: `http://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf`

- **What is a LSTM?** LSTM stands for Long Short Term Memory and it's a very popular type of RNN that is prominent for many AI implementations due to the architecture and benefits. If you visit the following lecture and section on RNN's (Annex 2), it will provide osme further context for LSTMs: `https://www.udemy.com/chatbot/learn/v4/t/lecture/8805327?start=0`

5.6 Lecture 40 - ChatBot - Step 21

- **I'm receiving the following error when running the code: AttributeError: module 'tensorflow.contrib.seq2seq' has no attribute 'prepare_attention'. How can I resolve it?** This is normally caused by the Tensorflow version. We are using version 1.0.0 and Python 3.5 for the first Chatbot version in the course. You can install Tensorflow 1.0.0 by running the following in the terminal or Anaconda prompt (if you are using a environment in Anaconda please activate it first) – install with pip install Tensorflow==1.0.0

5.7 Lecture 41 - ChatBot - Step 22

- **What is the difference between an encoder cell and encoder state?** An encoder cell is the cell inside the encoder RNN that contains the stacked LSTM layers. An encoder state is the output returned by the encoder RNN, right after the last fully connected layer.

5.8 Lecture 42 - ChatBot - Step 23

- **What is decoding_scope.reuse_variables() used for?** For our approach we get our test predictions with cross validations (that keeps 10% of the training data), and we also use this to predict our answers of the chatbot after it's training so we set our following variables and function with:

```
decoding_scope.reuse_variables()
    test_predictions = decode_test_set(encoder_state,
                                       decoder_cell,
                                       decoder_embeddings_matrix,
                                       word2int['<SOS>'],
                                       word2int['<EOS>'],
                                       sequence_length - 1,
                                       num_words,
                                       decoding_scope,
                                       output_function,
                                       keep_prob,
                                       batch_size)
return training_predictions, test_predictions
```

5.9 Lecture 43 - ChatBot - Step 24

- **What is the difference between the Decoder_embeddings_matrix and Decoder_embeddings_input? Also what is the purpose of using multiple columns in the decoder_embeddings_matrix?**
  The purpose of the embeddings_matrix is to compute more efficiently the embedding_input. Basically, you multiply your vector of inputs by the embeddings_matrix to get your embedded_inputs. For some further information please take a look at the following resources:

  https://www.tensorflow.org/versions/master/programmers_guide/embeddingandhttp://web.stanford.edu/class/cs20si/lectures/notes_04.pdf

5.10 Lecture 44 - Checkpoint!

6 Section 6: Part 3 – Training The Seq2Seq Model

6.1 Lecture 45 - What You'll Need For This Module

6.2 Lecture 46 - Welcome to Part 3 - Training the Seq2Seq Model

- **Why do we need to use the decode_training_set and decode_test_set? Also what role does attention_states have?** We need the decode_training_set to decode the encoded questions and answers of the training set (the second part of the Seq2Seq model) and we need the decode_test_set to decode the encoded questions answers of either the validation set, or simply new predictions that are not used anyway in the training. The role played by attention_states is to reinforce the Seq2Seq model with the attention process. You can take a look at the following scheme that highlights the process: http://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html#attention-decoder

6.3 Lecture 47 - ChatBot - Step 25

6.4 Lecture 48 - ChatBot - Step 26

6.5 Lecture 49 - ChatBot - Step 27

6.6 Lecture 50 - ChatBot - Step 28

6.7 Lecture 51 - ChatBot - Step 29

6.8 Lecture 52 - ChatBot - Step 30

- If you are running into any errors related to Error: seq2seq has no attribute prepare_attention please install Tensorflow 1.0.0 (if you are using a created environment please activate it first) with: pip install Tensorflow==1.0.0.

6.9 Lecture 53 - ChatBot - Step 31

- **Why are we using two learning rates?** The first creation of the learning rate in model_inputs was used in the function to create placeholders for inputs and targets since we are working with Tensorflow (essentially going from arrays to tensors for TF) we have to create the placeholder for it, for example: `https://www.tensorflow.org/api_docs/python/tf/placeholder`. When we are using the learning rate for training we are preparing the optimizer (the Adam optimizer) but we are using the learning rate that we already prepared in the hyper parameters(0.01).

- **I'm receiving the error that Seq2seq has no attribute prepare attention, how can I resolve it?** Please install Tensorflow v1.0.0 (if you are using a created environment please activate it first) with: pip install Tensorflow==1.0.0

6.10 Lecture 54 - ChatBot - Step 32

- **Why do the questions and answer sequences need to have the same fixed length?** That's because the answers are also questions. Indeed we get separately the questions and answers in Step 8, you notice that we only shift by one the couple (question, answer). So the answer to the previous question becomes the question to the next answer.

6.11 Lecture 55 - ChatBot - Step 33

- **What is the need of using start_index?** We need the start_index to set the first index of the question we are adding in the batch because we are dealing with a specific batch. So it's the first question / answer we are adding to the batch. To get it we are using batch_index * batch size due to starting at index 0. The first question added will have the start index of 0, overall in the loop of split_into_batches it's being used for that specific indexing.

6.12 Lecture 56 - ChatBot - Step 34

6.13 Lecture 57 - ChatBot - Step 35

- If you are having any issue on Windows with the checkpoint please us the following: checkpoint = "./chatbot_weights.ckpt"

6.14 Lecture 58 - ChatBot - Step 36

- **How can I train on GPU?** You will need to install Tensorflow with GPU, for example: `https://www.tensorflow.org/install/install_windows`

6.15 Lecture 59 - Checkpoint!

7 Section 7 Part 4 – Testing The Seq2Seq Model

7.1 Lecture 60 - What You'll Need For This Module

7.2 Lecture 61 - Welcome to Part 4 - Testing the Seq2Seq Model

7.3 Lecture 62 - ChatBot - Step 37

- **How many days will it take to train the Chatbot?** It will depend on the system but you can tell by running the code thanks to the "Training time on 100 batches" that will appear in the console. You need to multiply that training time by 20 (the number of batches) * 100 (the number of epochs).

7.4 Lecture 63 - ChatBot - Step 38

7.5 Lecture 64 - ChatBot - Step 39

- **During training the keep_prob was 0.5, but during validation the keep_prob was set to 1. Here in the live chat the keep_prob is again 0.5, but shouldn't it be 1?** All the neurons should be activated so the Dropout rate should be 0 and therefore keep_prob should be 1. I am saying "should" instead of "must" because it would also be okay to lower keep_prob in case you want to play with several teams of your neurons to predict new observations.

7.6 Lecture 65 - ChatBot - Step 40

7.7 Lecture 66 - Checkpoint!

8 Section 8 - ———- PART 5 - IMPROVING TUNING THE SEQ2SEQ MODEL ———-

8.1 Lecture 67 - ChatBot - Step 41: Improving Tuning the ChatBot

8.2 Lecture 68 - ChatBot - Step 42: Introduction to a new model setup

8.3 Lecture 69 - ChatBot - Step 43: Chatbot model discussion

8.4 Lecture 70 - ChatBot - Step 44: Tensorboard

8.5 Lecture 71 - ChatBot - Step 45: Run the new chatbot model

9 Section 9: Other ChatBot Implementations

9.1 Lecture 72 - What You'll Need For This Module

9.2 Lecture 73 - The Best ChatBot

- **I'm receiving an ImportError: No module named 'seq2seq_wrapper', how should I resolve it?** As a first step you can run either pip list or conda list in the terminal or anaconda prompt. This output will check the versions and you can look for the TF version. If TF 1.0.0 isn't installed, please install it with:

  pip install Tensorflow==1.0.0

9.3 Lecture 74 - A ChatBot Implementation in TensorFlow 1.4

9.4 Lecture 75 - A ChatBot Implementation in PyTorch

10 Section 10 Annex 1: Artificial Neural Networks

10.1 Lecture 76 - Plan of Attack

10.2 Lecture 77 - The Neuron

10.3 Lecture 78 - The Activation Function

10.4 Lecture 79 - How do Neural Networks work?

10.5 Lecture 80 - How do Neural Networks learn?

10.6 Lecture 81 - Gradient Descent

10.7 Lecture 82 - Stochastic Gradient Descent

10.8 Lecture 83 - Backpropagation

- **What is the difference between stochastic gradient descent and backpropagation?** Backpropagation is the algorithm that is used to calculate the gradient of the loss function with respect to parameters of the neural network. Gradient descent is the optimization algorithm that is used to find parameters that minimize the loss function"

  `www.quora.com/What-is-the-difference-between-gradient-descent-and-back-propagation-in-deep-learning-Are-they`

11 Section 11 - Annex 2: Recurrent Neural Networks

11.1 Lecture 84 - Plan of Attack

11.2 Lecture 85 - What are Recurrent Neural Networks?

11.3 Lecture 86 - Vanishing Gradient Problems for RNNs

11.4 Lecture 87 - Long Short Term Memory

11.5 Lecture 88 - Practical Intuition

11.6 Lecture 89 - Long Short Term Memory Variations

12 FAQ Final Notes

Thank you to all of the students who have helped contribute toward this FAQ by discussing elements of the course and sharing information in the Q&A. We will revise the document as the FAQ is updated so please keep an eye out for any future announcements.