UDACITY

Logout

PROJECT

Recurrent Neural Networks

A part of the Artificial Intelligence Nanodegree Program

PROJECT REVIEW | CODE REVIEW | NOTES

Requires Changes

SHARE YOUR ACCOMPLISHMENT

1 SPECIFICATION REQUIRES CHANGES

I really liked reviewing your project. Everything apart from one tiny bit of removing the non-English characters works perfectly. Please consider making the suggested changes and resubmit. Good luck :)

Files Submitted

The submission includes all required file RNN_project_student_version.ipynb All code must be written ONLY in the TODO sections and no previous code should be modified.	Rate this review
Great! The submission includes the required files and all the TODO sections have been completed.	

Step 1: Implement a function to window time series

The submission returns the proper windowed version of input time series of proper dimension listed in the notebook.

Awesome! The time series output looks really good.
Here is a link to test your code on some other time series: https://datamarket.com/data/list/?q=provider%3Atsdl

Step 2: Create a simple RNN model for regression

The submission constructs an RNN model in keras with LSTM module of dimension defined in the notebook.

Good job! I am adding a link to Keras RNN documentation if you would like to read more about it: https://keras.io/layers/recurrent/

Step 3: Clean up a large text corpus

The submission removes all non-english / non-punctuation characters. (English characters should include string.ascii_lowercase and punctuation includes [' ', '!', ',', '.', ':', ';', '?'] (space, eclamation mark, comma, period, colon, semicolon, question mark))

You removed most of the non-English / non-punctuation characters. However, on running your notebook I found some remaining.

Note: There should be 26 letters and 7 punctuation, for a total of 33 characters. English characters should include string.ascii_lowercase and punctuation includes [' ', '!', ',', '.', ':', ';', '?'] (space, eclamation mark, comma, period, colon, semicolon, question mark)

Rate this review

Step 4: Implement a function to window a large text corpus

The submission returns the proper windowed version of input text of proper dimension listed in the notebook.

Perfect! A proper windowed version of the input text is generated.

Step 5: Create an RNN perform multiclass

The submission constructs an RNN model in keras with LSTM module of dimension defined in the notebook.

Good job with the RNN model and explaining the input parameters.

Step 6: Generate text using a fully trained RNN

The submission presents examples of generated text from a trained RNN module. The majority of this generated text should consist of real english words.

Awesome! The majority of the generated text is English.

RESUBMIT PROJECT

DOWNLOAD PROJECT

Rate this review

Best practices for your project resubmission

Rate this review

Ben shares 5 helpful tips to get you through revising and resubmitting your project.