UDACITY

PROJECT

Recurrent Neural Networks

A part of the Artificial Intelligence Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

Kudos ! I think you've done a perfect job of implementing a recurrent neural net fully. It's very clear that you have a good understanding of the basics. Keep improving and keep learning.

As it appears, you have some idea of LSTMs & RNNs, here's a very popular blog that might help you in visually understanding further details.

Advanced tips for improving net results

- Try and use deeper architectures, which have general tendency to blow up or vanish the gradients - so there's a net architecture known as Residual Nets, used to circumnavigate the issues with deeper architectures
- Try using more fully connected layers or Bi-Directional LSTMs or GRUs to make the predictions even better
- Try and use more sophisticated methods like `lemmatisation` and `stemming` to create a more pruned vocabulary. Have a look at the NLTK library to understand more operations

If you are keen on learning a bit more into what Natural Language Scientists use regularly in their nets. Try reading up a bit more on

- Word2Vec Algorithm
- Glove Algorithm
- Sequence2Sequence tutorial

Keep up the good work !

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.

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.

Correct!

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.

Correct!

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, exclamation mark, comma, period, colon, semicolon, question mark))

Correct!

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.

Correct!

Step 5: Create an RNN perform multiclass

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

Correct!

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.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

Student FAQ