UDACITY

Back to Artificial Intelligence Nanodegree and Specializations

Recurrent Neural Networks

REVIEW

CODE REVIEW 2

HISTORY

▼ my_answers.py 2

```
import numpy as np
import string

import keras
from keras.layers import Activation, Dense, LSTM
from keras.models import Sequential



# TODO: fill out the function below that transforms the input series
# and window-size into a set of input/output pairs for use with our RNN model
def window_transform_series(series, window_size):
    # containers for input/output pairs
    X = []
    y = []

    for i in range(0, len(series)-window_size):
        X.append(series[i:i+window_size])
        y.append(series[i+window_size])

    # reshape each
    X = np.asarray(X)
    X.shape = (np.shape(X)[0:2])
    y = np.asarray(y)
    y.shape = (len(y),1)

    return X,y

```

```
29
30 # TODO: build an RNN to perform regression on our time series input/output data
31 def build_part1_RNN(window_size):
32     model = Sequential()
33     model.add(LSTM(5, input_shape=(window_size, 1)))
34     model.add(Dense(1))
35
36     return model
37
38
39 ### TODO: return the text input with only ascii lowercase and the punctuation
40 def cleaned_text(text):
41     punctuation = ['!', ',', '.', ':', ';', '?']
42
43     # combine alphabet chars and allowed punctuation
44     a2z = list(string.ascii_lowercase)
45     ok_chars = set(punctuation + a2z + [' '])
46
47     # identify chars to remove and replace them
48     all_chars = set(text)
49     chars_to_remove = all_chars - ok_chars
50
51     for c in chars_to_remove:
52         text = text.replace(c, ' ')
```

AWESOME

Simple, easy to understand way to do this.

```
53
54     print("{} characters removed from text.".format(len(chars_to_remove)))
55     print(chars_to_remove)
56
57     return text
58
59
60 ### TODO: fill out the function below that transforms the input text and windo
61 def window_transform_text(text, window_size, step_size):
62     # containers for input/output pairs
63     inputs = []
64     outputs = []
65
66     i = 0
67     while i < len(text) - window_size:
68         inputs.append(text[i:i + window_size])
69         outputs.append(text[i + window_size])
70         i += step_size
```

SUGGESTION

For future reference, take a look at the other optional parameters that `range` has.

`range(start, stop[, step])`

https://docs.python.org/3/library/functions.html#func-range

```
71
72      return inputs, outputs
73
74
75  # TODO build the required RNN model:
76  # a single LSTM hidden layer with softmax activation, categorical_crossentropy
77  def build_part2_RNN(window_size, num_chars):
78      model = Sequential()
79      model.add(LSTM(200, input_shape=(window_size, num_chars)))
80      model.add(Dense(num_chars))
81      model.add(Activation('softmax'))
82
83      return model
```

RETURN TO PATH

Rate this review

Student FAQ