Assignment 10-1

May 20, 2021

0.1 Create a tokenize function

[1]:
```
import string
```

[2]:
```
def tokenize(sentence):
    # Split the sentence by spaces
    words = sentence.split()
    # Remove punctuation
    table = str.maketrans('', '', string.punctuation)
    stripped = [w.translate(table) for w in words]
    return stripped
```

[3]:
```
sentence = "This is my sentence, to parse. Get all punctuation out# of here!"
tokens = tokenize(sentence)
print(type(tokens))
print(tokens)
```

```
<class 'list'>
['This', 'is', 'my', 'sentence', 'to', 'parse', 'Get', 'all', 'punctuation',
'out', 'of', 'here']
```

Assignment 10.1 b

[4]:
```
import nltk
```

[5]:
```
def ngram(paragraph, n):
    # Split the sentence by spaces
    words = paragraph.split()
    # Remove punctuation
    table = str.maketrans('', '', string.punctuation)
    stripped = [w.translate(table) for w in words]
    bi_grams = nltk.ngrams(stripped, n)
    return bi_grams
```

[6]:
```
paragraph = "This is my sentence, to parse. Get all punctuation out# of here!"
bi_grams = ngram(paragraph, 3)
for gram in bi_grams:
    print(gram)
```

```
('This', 'is', 'my')
('is', 'my', 'sentence')
('my', 'sentence', 'to')
('sentence', 'to', 'parse')
('to', 'parse', 'Get')
('parse', 'Get', 'all')
('Get', 'all', 'punctuation')
('all', 'punctuation', 'out')
('punctuation', 'out', 'of')
('out', 'of', 'here')
```

Assignment 10.1 c - Create a Vector

[7]:
```
import string
import nltk
from numpy import array
from numpy import argmax
from keras.utils import to_categorical
```

[8]:
```
def onehtencode(data):
    data = array(data)
    print("Received array")
    print(data)
    # one hot encode
    encoded = to_categorical(data)
    return encoded
```

[9]:
```
data = [1, 3, 2, 0, 3, 2, 2, 1, 0, 1]
encodedval = onehtencode(data)
print("One Hot Encoded values")
print(encodedval)
```

```
Received array
[1 3 2 0 3 2 2 1 0 1]
One Hot Encoded values
[[0. 1. 0. 0.]
 [0. 0. 0. 1.]
 [0. 0. 1. 0.]
 [1. 0. 0. 0.]
 [0. 0. 0. 1.]
 [0. 0. 1. 0.]
 [0. 0. 1. 0.]
 [0. 1. 0. 0.]
 [1. 0. 0. 0.]
 [0. 1. 0. 0.]]
```

[ ]: