In [6]:

```
# RNN Python example code

import tensorflow as tf
import numpy as np
i = 4
s = 5
n = 1
inputs = np.random.random([i, s, n]).astype(np.float32)
simple_rnn = tf.keras.layers.SimpleRNN(i)
print(inputs)
output = simple_rnn(inputs) # The output has shape `[32, 4]`.
print(output)
```

```
[[[0.6803922 ]
  [0.5748543 ]
  [0.81096804]
  [0.81913304]
  [0.95446867]]

 [[0.7883225 ]
  [0.67579377]
  [0.4745802 ]
  [0.29190052]
  [0.84892696]]

 [[0.03816856]
  [0.92490965]
  [0.09551006]
  [0.9884428 ]
  [0.00197868]]

 [[0.36617085]
  [0.31019738]
  [0.5009294 ]
  [0.6544853 ]
  [0.02470805]]]
tf.Tensor(
[[ 0.9341498  -0.26840273  0.43220943 -0.9447197 ]
 [ 0.9043899  -0.38562635  0.46590722 -0.8718167 ]
 [ 0.6701342  -0.5873792   0.16694997 -0.6722176 ]
 [ 0.6472012  -0.60866165  0.32480136 -0.6002206 ]], shape=(4, 4), dtype=flo
at32)
```

In [7]:

```
# Another example of Keras SimpleRNN, showing how to create an RNN model and show the model

from keras.models import Sequential
from keras.layers import Dense, SimpleRNN, Activation
from keras import optimizers
model = Sequential()
model.add(SimpleRNN(50, input_shape = (49,1), return_sequences =
False))
model.add(Dense(46))
model.add(Activation('softmax'))

adam = optimizers.Adam(lr = 0.001)
model.compile(loss = 'categorical_crossentropy', optimizer = adam,
metrics = ['accuracy'])
print(model.summary())
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 simple_rnn_1 (SimpleRNN)    (None, 50)                2600

 dense (Dense)               (None, 46)                2346

 activation (Activation)     (None, 46)                0

=================================================================
Total params: 4,946
Trainable params: 4,946
Non-trainable params: 0
_________________________________________________________________
None


c:\pythonn\lib\site-packages\keras\optimizers\optimizer_v2\adam.py:110: User
Warning: The `lr` argument is deprecated, use `learning_rate` instead.
  super(Adam, self).__init__(name, **kwargs)
```

In [8]:

```
# A Python code that shows how to create a simple Keras LSTM network.

from tensorflow import keras
from tensorflow.keras import layers
model = keras.Sequential()
model.add(layers.Embedding(input_dim=100, output_dim=32))
model.add(layers.LSTM(64))
model.add(layers.Dense(1))
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, None, 32)          3200

 lstm (LSTM)                 (None, 64)                24832

 dense_1 (Dense)             (None, 1)                 65

=================================================================
Total params: 28,097
Trainable params: 28,097
Non-trainable params: 0
_________________________________________________________________
```

In [10]:

```
# A Python example code for using the NLTK library for
# text analysis. The text in the code is from the Wikipedia page “Artificial Intelligence”

import nltk
nltk.download('averaged_perceptron_tagger')
nltk.download('maxent_ne_chunker')
nltk.download('words')
sentence = """
Artificial intelligence (AI), is intelligence demonstrated by
machines,
 unlike the natural intelligence displayed by humans and animals.
 Leading AI textbooks define the field as the study of "intelligent
agents":
 any device that perceives its environment and takes actions that
maximize its
 chance of successfully achieving its goals.[3] Colloquially, the
term
 "artificial intelligence" is often used to describe machines (or
computers)
 that mimic "cognitive" functions that humans associate with the
human mind,
 such as "learning" and "problem solving".[4]
"""
tokens = nltk.word_tokenize(sentence)
print(tokens)
tagged = nltk.pos_tag(tokens)
print(tagged)
entities = nltk.chunk.ne_chunk(tagged)
print(entities)
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger.zip.
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]   Package maxent_ne_chunker is already up-to-date!
[nltk_data] Downloading package words to
[nltk_data]     C:\Users\hp5cd\AppData\Roaming\nltk_data...
[nltk_data]   Package words is already up-to-date!
```