How to Develop Word-Based Neural Language Models in Python with Keras

Language modeling involves predicting the next word in a sequence given the sequence of words already present.
The choice of how the language model is framed must match how the language model is intended to be used.

In this tutorial, you will discover how the framing of a language model affects the skill of the model when generating short sequences from a nursery rhyme.

Framing Language Modeling

Jack and Jill went up the hill
To fetch a pail of water
Jack fell down and broke his crown
And Jill came tumbling after

A statistical language model is learned from raw text and predicts the probability of the next word in the sequence given the words already present in the sequence.
They can also be developed as standalone models and used for generating new sequences that have the same statistical properties as the source text.

Language models both learn and predict one word at a time. The training of the network involves providing sequences of words as input that are processed one at a time.

There is no single best approach, just different framings that may suit different applications.

Methods for text sequences

- **One-Word-In, One-Word-Out Sequences**
- **Line-by-Line Sequence**
- **Two-Words-In, One-Word-Out Sequence**

Model 1: One-Word-In, One-Word-Out Sequences

Given one word as input, the model will learn to predict the next word in the sequence.

In [1]:

```
from numpy import array
from keras.preprocessing.text import Tokenizer # for encoding our text
from keras.utils import to_categorical
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Embedding
```

```
Using TensorFlow backend.
```

The first step is to encode the text as integers.(*similar to variable encoding*)
Keras provides the Tokenizer (https://keras.io/preprocessing/text/#tokenizer) class that can be used to perform this encoding.

In [2]:

```
# source text
data = """ Jack and Jill went up the hill\n
        To fetch a pail of water\n
        Jack fell down and broke his crown\n
        And Jill came tumbling after\n """
```

First, the Tokenizer is to `fit` on the source text to develop the mapping from words to unique integers. Then sequences of text can be converted to sequences of integers by calling the `texts_to_sequences()` function.

In [3]:

```
# integer encode text
tokenizer = Tokenizer()
tokenizer.fit_on_texts([data])
encoded = tokenizer.texts_to_sequences([data])[0]
```

We will need to know the size of the vocabulary later for both defining the word embedding layer in the model, and for encoding output words using a one hot encoding.
The size of the vocabulary can be retrieved from the trained Tokenizer by accessing the `word_index` attribute.

In [4]:

```
# determine the vocabulary size
vocab_size = len(tokenizer.word_index) + 1
print('Vocabulary Size: %d' % vocab_size)
```

```
Vocabulary Size: 22
```

Next, we need to create sequences of words to fit the model with one word as input and one word as output.

In [5]:

```
# create word -> word sequences
sequences = list()
for i in range(1, len(encoded)):
    sequence = encoded[i-1:i+1]
    sequences.append(sequence)
print('Total Sequences: %d' % len(sequences))
```

```
Total Sequences: 24
```

We can then split the sequences into input (X) and output elements (y). This is straightforward as we only have two columns in the data.

In [6]:

```
# split into X and y elements
sequences = array(sequences)
X, y = sequences[:,0],sequences[:,1]
```

We will fit our model to predict a probability distribution across all words in the vocabulary. That means that we need to turn the output element from a single integer into a one hot encoding with a 0 for every word in the vocabulary and a 1 for the actual word that the value. This gives the network a ground truth to aim for from which we can calculate error and update the model.

Keras provides the `to_categorical()` function that we can use to convert the integer to a one hot encoding while specifying the number of classes as the vocabulary size.

In [7]:

```
# one hot encode outputs
y = to_categorical(y, num_classes=vocab_size)
```

Making our artificial neural network

In [8]:

```
# define model
model = Sequential()
model.add(Embedding(vocab_size, 10, input_length=1))
model.add(LSTM(50))
model.add(Dense(vocab_size, activation='softmax'))
print(model.summary())
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_1 (Embedding)      (None, 1, 10)             220       
_________________________________________________________________
lstm_1 (LSTM)                (None, 50)                12200     
_________________________________________________________________
dense_1 (Dense)              (None, 22)                1122      
=================================================================
Total params: 13,542
Trainable params: 13,542
Non-trainable params: 0
_________________________________________________________________
None
```

In [9]:

```
# compile network
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

In [10]:

```
# fit network
model.fit(X, y, epochs=0, verbose=1)
```

Out[10]:

```
<keras.callbacks.History at 0x2e6d7540400>
```

After the model is fit, we test it by passing it a given word from the vocabulary and having the model predict the next word. Here we pass in ‘Jack‘ by encoding it and calling `model.predict_classes()` to get the integer output for the predicted word. This is then looked up in the vocabulary mapping to give the associated word.

In [11]:

```
# evaluate
in_text = 'Jack'
print(in_text)
```

```
Jack
```

In [12]:

```
encoded = tokenizer.texts_to_sequences([in_text])[0]
encoded = array(encoded)
yhat = model.predict_classes(encoded, verbose=0)
for word, index in tokenizer.word_index.items():
    if index == yhat:
        print(word)
```

```
went
```

In [13]:

```
# generate a sequence from the model
def generate_seq(model, tokenizer, seed_text, n_words):
    in_text, result = seed_text, seed_text
    # generate a fixed number of words
    for _ in range(n_words):
        # encode the text as integer
        encoded = tokenizer.texts_to_sequences([in_text])[0]
        encoded = array(encoded)
        # predict a word in the vocabulary
        yhat = model.predict_classes(encoded, verbose=0)
        # map predicted word index to word
        out_word = ''
        for word, index in tokenizer.word_index.items():
            if index == yhat:
                out_word = word
                break
        # append to input
        in_text, result = out_word, result + ' ' + out_word
    return result
```

In [14]:

```
# evaluate
print(generate_seq(model, tokenizer, 'Jack', 6))
```

```
Jack went went went went went went
```

Model 2: Line-by-Line Sequence

Another approach is to split up the source text line-by-line, then break each line down into a series of words that build up.

This approach may allow the model to use the context of each line to help the model in those cases where a simple one-word-in-and-out model creates ambiguity.

In this case, this comes at the cost of predicting words across lines, which might be fine for now if we are only interested in modeling and generating lines of text.

Note that in this representation, we will require a padding of sequences to ensure they meet a fixed length input. This is a requirement when using Keras.

In [15]:

```
from numpy import array
from keras.preprocessing.text import Tokenizer
from keras.utils import to_categorical
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Embedding
```

In [16]:

```
# source text
data = """ Jack and Jill went up the hill\n
        To fetch a pail of water\n
        Jack fell down and broke his crown\n
        And Jill came tumbling after\n """
```

In [17]:

```
# prepare the tokenizer on the source text
tokenizer = Tokenizer()
tokenizer.fit_on_texts([data])
```

In [18]:

```
# determine the vocabulary size
vocab_size = len(tokenizer.word_index) + 1
print('Vocabulary Size: %d' % vocab_size)
```

```
Vocabulary Size: 22
```

First, we can create the sequences of integers, line-by-line by using the Tokenizer already fit on the source text.

In [19]:

```
# create line-based sequences
sequences = list()
for line in data.split('\n'):
    encoded = tokenizer.texts_to_sequences([line])[0]
    for i in range(1, len(encoded)):
        sequence = encoded[:i+1]
        sequences.append(sequence)
print('Total Sequences: %d' % len(sequences))
```

```
Total Sequences: 21
```

Next, we can pad the prepared sequences. We can do this using the `pad_sequences()` function provided in Keras. This first involves finding the longest sequence, then using that as the length by which to pad-out all other sequences.

In [20]:

```
# pad input sequences
max_length = max([len(seq) for seq in sequences])
sequences = pad_sequences(sequences, maxlen=max_length, padding='pre')
print('Max Sequence Length: %d' % max_length)
```

```
Max Sequence Length: 7
```

Next, we can split the sequences into input and output elements, much like before.

In [21]:

```
# split into input and output elements
sequences = array(sequences)
X, y = sequences[:,:-1],sequences[:,-1]
y = to_categorical(y, num_classes=vocab_size)
```

The model can then be defined as before, except the input sequences are now longer than a single word. Specifically, they are max_length-1 in length, -1 because when we calculated the maximum length of sequences, they included the input and output elements.

In [22]:

```
# define model
model = Sequential()
model.add(Embedding(vocab_size, 10, input_length=max_length-1))
model.add(LSTM(50))
model.add(Dense(vocab_size, activation='softmax'))
print(model.summary())
```

Layer (type)	Output Shape	Param #
embedding_2 (Embedding)	(None, 6, 10)	220
lstm_2 (LSTM)	(None, 50)	12200
dense_2 (Dense)	(None, 22)	1122

```
Total params: 13,542
Trainable params: 13,542
Non-trainable params: 0
```

```
None
```

In [23]:

```
# compile network
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

In [24]:

```
# fit network
model.fit(X, y, epochs=500, verbose=2)
```

```
Epoch 1/500
 - 1s - loss: 3.0912 - acc: 0.0000e+00
Epoch 2/500
 - 0s - loss: 3.0898 - acc: 0.0476
Epoch 3/500
 - 0s - loss: 3.0886 - acc: 0.0952
Epoch 4/500
 - 0s - loss: 3.0871 - acc: 0.0952
Epoch 5/500
 - 0s - loss: 3.0857 - acc: 0.0952
Epoch 6/500
 - 0s - loss: 3.0842 - acc: 0.0952
Epoch 7/500
 - 0s - loss: 3.0827 - acc: 0.0952
Epoch 8/500
 - 0s - loss: 3.0812 - acc: 0.0952
Epoch 9/500
 - 0s - loss: 3.0796 - acc: 0.0952
Epoch 10/500
 - 0s - loss: 3.0780 - acc: 0.0952
Epoch 11/500
 - 0s - loss: 3.0762 - acc: 0.0952
Epoch 12/500
 - 0s - loss: 3.0744 - acc: 0.0952
Epoch 13/500
 - 0s - loss: 3.0725 - acc: 0.0952
Epoch 14/500
 - 0s - loss: 3.0705 - acc: 0.0952
Epoch 15/500
 - 0s - loss: 3.0683 - acc: 0.0952
Epoch 16/500
 - 0s - loss: 3.0661 - acc: 0.0952
Epoch 17/500
 - 0s - loss: 3.0637 - acc: 0.0952
Epoch 18/500
 - 0s - loss: 3.0612 - acc: 0.0952
Epoch 19/500
 - 0s - loss: 3.0585 - acc: 0.0952
Epoch 20/500
 - 0s - loss: 3.0557 - acc: 0.0952
Epoch 21/500
 - 0s - loss: 3.0526 - acc: 0.0952
Epoch 22/500
 - 0s - loss: 3.0493 - acc: 0.0952
Epoch 23/500
 - 0s - loss: 3.0458 - acc: 0.0952
Epoch 24/500
 - 0s - loss: 3.0421 - acc: 0.0952
Epoch 25/500
 - 0s - loss: 3.0381 - acc: 0.0952
Epoch 26/500
 - 0s - loss: 3.0338 - acc: 0.0952
Epoch 27/500
 - 0s - loss: 3.0291 - acc: 0.0952
Epoch 28/500
 - 0s - loss: 3.0241 - acc: 0.0952
Epoch 29/500
 - 0s - loss: 3.0187 - acc: 0.0952
Epoch 30/500
 - 0s - loss: 3.0129 - acc: 0.0952
Epoch 31/500
```

```
 - 0s - loss: 3.0067 - acc: 0.0952
Epoch 32/500
 - 0s - loss: 3.0001 - acc: 0.0952
Epoch 33/500
 - 0s - loss: 2.9929 - acc: 0.0952
Epoch 34/500
 - 0s - loss: 2.9851 - acc: 0.0952
Epoch 35/500
 - 0s - loss: 2.9770 - acc: 0.0952
Epoch 36/500
 - 0s - loss: 2.9684 - acc: 0.0952
Epoch 37/500
 - 0s - loss: 2.9594 - acc: 0.0952
Epoch 38/500
 - 0s - loss: 2.9500 - acc: 0.0952
Epoch 39/500
 - 0s - loss: 2.9403 - acc: 0.0952
Epoch 40/500
 - 0s - loss: 2.9305 - acc: 0.0952
Epoch 41/500
 - 0s - loss: 2.9208 - acc: 0.0952
Epoch 42/500
 - 0s - loss: 2.9114 - acc: 0.0952
Epoch 43/500
 - 0s - loss: 2.9026 - acc: 0.0952
Epoch 44/500
 - 0s - loss: 2.8945 - acc: 0.0952
Epoch 45/500
 - 0s - loss: 2.8870 - acc: 0.0952
Epoch 46/500
 - 0s - loss: 2.8800 - acc: 0.0952
Epoch 47/500
 - 0s - loss: 2.8728 - acc: 0.0952
Epoch 48/500
 - 0s - loss: 2.8651 - acc: 0.0952
Epoch 49/500
 - 0s - loss: 2.8563 - acc: 0.0952
Epoch 50/500
 - 0s - loss: 2.8465 - acc: 0.0952
Epoch 51/500
 - 0s - loss: 2.8358 - acc: 0.0952
Epoch 52/500
 - 0s - loss: 2.8243 - acc: 0.0952
Epoch 53/500
 - 0s - loss: 2.8125 - acc: 0.1429
Epoch 54/500
 - 0s - loss: 2.8006 - acc: 0.1429
Epoch 55/500
 - 0s - loss: 2.7885 - acc: 0.1429
Epoch 56/500
 - 0s - loss: 2.7763 - acc: 0.1429
Epoch 57/500
 - 0s - loss: 2.7637 - acc: 0.1429
Epoch 58/500
 - 0s - loss: 2.7506 - acc: 0.1429
Epoch 59/500
 - 0s - loss: 2.7367 - acc: 0.1429
Epoch 60/500
 - 0s - loss: 2.7220 - acc: 0.1429
Epoch 61/500
 - 0s - loss: 2.7063 - acc: 0.1429
```

```
Epoch 62/500
 - 0s - loss: 2.6895 - acc: 0.1429
Epoch 63/500
 - 0s - loss: 2.6721 - acc: 0.1429
Epoch 64/500
 - 0s - loss: 2.6540 - acc: 0.1429
Epoch 65/500
 - 0s - loss: 2.6353 - acc: 0.1429
Epoch 66/500
 - 0s - loss: 2.6158 - acc: 0.1429
Epoch 67/500
 - 0s - loss: 2.5954 - acc: 0.1429
Epoch 68/500
 - 0s - loss: 2.5740 - acc: 0.1429
Epoch 69/500
 - 0s - loss: 2.5515 - acc: 0.1429
Epoch 70/500
 - 0s - loss: 2.5278 - acc: 0.1429
Epoch 71/500
 - 0s - loss: 2.5030 - acc: 0.1429
Epoch 72/500
 - 0s - loss: 2.4777 - acc: 0.1429
Epoch 73/500
 - 0s - loss: 2.4517 - acc: 0.1429
Epoch 74/500
 - 0s - loss: 2.4246 - acc: 0.2381
Epoch 75/500
 - 0s - loss: 2.3967 - acc: 0.2857
Epoch 76/500
 - 0s - loss: 2.3685 - acc: 0.2857
Epoch 77/500
 - 0s - loss: 2.3397 - acc: 0.2857
Epoch 78/500
 - 0s - loss: 2.3111 - acc: 0.2857
Epoch 79/500
 - 0s - loss: 2.2825 - acc: 0.3333
Epoch 80/500
 - 0s - loss: 2.2542 - acc: 0.3333
Epoch 81/500
 - 0s - loss: 2.2261 - acc: 0.3810
Epoch 82/500
 - 0s - loss: 2.1990 - acc: 0.3810
Epoch 83/500
 - 0s - loss: 2.1725 - acc: 0.3810
Epoch 84/500
 - 0s - loss: 2.1456 - acc: 0.3810
Epoch 85/500
 - 0s - loss: 2.1182 - acc: 0.3810
Epoch 86/500
 - 0s - loss: 2.0911 - acc: 0.4286
Epoch 87/500
 - 0s - loss: 2.0650 - acc: 0.4286
Epoch 88/500
 - 0s - loss: 2.0395 - acc: 0.4762
Epoch 89/500
 - 0s - loss: 2.0141 - acc: 0.5238
Epoch 90/500
 - 0s - loss: 1.9889 - acc: 0.5238
Epoch 91/500
 - 0s - loss: 1.9639 - acc: 0.5714
Epoch 92/500
```

```
 - 0s - loss: 1.9397 - acc: 0.5714
Epoch 93/500
 - 0s - loss: 1.9155 - acc: 0.5714
Epoch 94/500
 - 0s - loss: 1.8916 - acc: 0.5714
Epoch 95/500
 - 0s - loss: 1.8681 - acc: 0.5714
Epoch 96/500
 - 0s - loss: 1.8450 - acc: 0.5714
Epoch 97/500
 - 0s - loss: 1.8224 - acc: 0.6190
Epoch 98/500
 - 0s - loss: 1.7998 - acc: 0.6667
Epoch 99/500
 - 0s - loss: 1.7773 - acc: 0.6667
Epoch 100/500
 - 0s - loss: 1.7547 - acc: 0.6667
Epoch 101/500
 - 0s - loss: 1.7320 - acc: 0.7143
Epoch 102/500
 - 0s - loss: 1.7092 - acc: 0.7143
Epoch 103/500
 - 0s - loss: 1.6867 - acc: 0.7143
Epoch 104/500
 - 0s - loss: 1.6653 - acc: 0.7143
Epoch 105/500
 - 0s - loss: 1.6450 - acc: 0.7143
Epoch 106/500
 - 0s - loss: 1.6249 - acc: 0.7143
Epoch 107/500
 - 0s - loss: 1.6044 - acc: 0.7143
Epoch 108/500
 - 0s - loss: 1.5836 - acc: 0.7143
Epoch 109/500
 - 0s - loss: 1.5627 - acc: 0.7143
Epoch 110/500
 - 0s - loss: 1.5421 - acc: 0.7143
Epoch 111/500
 - 0s - loss: 1.5219 - acc: 0.7143
Epoch 112/500
 - 0s - loss: 1.5021 - acc: 0.7143
Epoch 113/500
 - 0s - loss: 1.4828 - acc: 0.7143
Epoch 114/500
 - 0s - loss: 1.4638 - acc: 0.7143
Epoch 115/500
 - 0s - loss: 1.4449 - acc: 0.7143
Epoch 116/500
 - 0s - loss: 1.4264 - acc: 0.7143
Epoch 117/500
 - 0s - loss: 1.4080 - acc: 0.7143
Epoch 118/500
 - 0s - loss: 1.3894 - acc: 0.7143
Epoch 119/500
 - 0s - loss: 1.3708 - acc: 0.7143
Epoch 120/500
 - 0s - loss: 1.3522 - acc: 0.7143
Epoch 121/500
 - 0s - loss: 1.3339 - acc: 0.7143
Epoch 122/500
 - 0s - loss: 1.3160 - acc: 0.7143
```

```
Epoch 123/500
 - 0s - loss: 1.2982 - acc: 0.7143
Epoch 124/500
 - 0s - loss: 1.2805 - acc: 0.7143
Epoch 125/500
 - 0s - loss: 1.2630 - acc: 0.7143
Epoch 126/500
 - 0s - loss: 1.2455 - acc: 0.7143
Epoch 127/500
 - 0s - loss: 1.2280 - acc: 0.7143
Epoch 128/500
 - 0s - loss: 1.2107 - acc: 0.7143
Epoch 129/500
 - 0s - loss: 1.1936 - acc: 0.7143
Epoch 130/500
 - 0s - loss: 1.1769 - acc: 0.7143
Epoch 131/500
 - 0s - loss: 1.1605 - acc: 0.7143
Epoch 132/500
 - 0s - loss: 1.1445 - acc: 0.7143
Epoch 133/500
 - 0s - loss: 1.1288 - acc: 0.7143
Epoch 134/500
 - 0s - loss: 1.1133 - acc: 0.7143
Epoch 135/500
 - 0s - loss: 1.0981 - acc: 0.7619
Epoch 136/500
 - 0s - loss: 1.0833 - acc: 0.7619
Epoch 137/500
 - 0s - loss: 1.0687 - acc: 0.7619
Epoch 138/500
 - 0s - loss: 1.0545 - acc: 0.7619
Epoch 139/500
 - 0s - loss: 1.0406 - acc: 0.7619
Epoch 140/500
 - 0s - loss: 1.0269 - acc: 0.7619
Epoch 141/500
 - 0s - loss: 1.0134 - acc: 0.7619
Epoch 142/500
 - 0s - loss: 1.0003 - acc: 0.7619
Epoch 143/500
 - 0s - loss: 0.9875 - acc: 0.8095
Epoch 144/500
 - 0s - loss: 0.9751 - acc: 0.7619
Epoch 145/500
 - 0s - loss: 0.9629 - acc: 0.7619
Epoch 146/500
 - 0s - loss: 0.9510 - acc: 0.8095
Epoch 147/500
 - 0s - loss: 0.9395 - acc: 0.8095
Epoch 148/500
 - 0s - loss: 0.9282 - acc: 0.8095
Epoch 149/500
 - 0s - loss: 0.9172 - acc: 0.8095
Epoch 150/500
 - 0s - loss: 0.9064 - acc: 0.8095
Epoch 151/500
 - 0s - loss: 0.8960 - acc: 0.8095
Epoch 152/500
 - 0s - loss: 0.8858 - acc: 0.8095
Epoch 153/500
```

```
 - 0s - loss: 0.8760 - acc: 0.8095
Epoch 154/500
 - 0s - loss: 0.8664 - acc: 0.8095
Epoch 155/500
 - 0s - loss: 0.8571 - acc: 0.8095
Epoch 156/500
 - 0s - loss: 0.8480 - acc: 0.8095
Epoch 157/500
 - 0s - loss: 0.8391 - acc: 0.8095
Epoch 158/500
 - 0s - loss: 0.8303 - acc: 0.8095
Epoch 159/500
 - 0s - loss: 0.8218 - acc: 0.8095
Epoch 160/500
 - 0s - loss: 0.8135 - acc: 0.8095
Epoch 161/500
 - 0s - loss: 0.8055 - acc: 0.8095
Epoch 162/500
 - 0s - loss: 0.7976 - acc: 0.8095
Epoch 163/500
 - 0s - loss: 0.7900 - acc: 0.8095
Epoch 164/500
 - 0s - loss: 0.7825 - acc: 0.8095
Epoch 165/500
 - 0s - loss: 0.7752 - acc: 0.8095
Epoch 166/500
 - 0s - loss: 0.7681 - acc: 0.8095
Epoch 167/500
 - 0s - loss: 0.7611 - acc: 0.8095
Epoch 168/500
 - 0s - loss: 0.7542 - acc: 0.8095
Epoch 169/500
 - 0s - loss: 0.7475 - acc: 0.8095
Epoch 170/500
 - 0s - loss: 0.7410 - acc: 0.8095
Epoch 171/500
 - 0s - loss: 0.7346 - acc: 0.8095
Epoch 172/500
 - 0s - loss: 0.7283 - acc: 0.8095
Epoch 173/500
 - 0s - loss: 0.7221 - acc: 0.8095
Epoch 174/500
 - 0s - loss: 0.7161 - acc: 0.8095
Epoch 175/500
 - 0s - loss: 0.7101 - acc: 0.8095
Epoch 176/500
 - 0s - loss: 0.7043 - acc: 0.8095
Epoch 177/500
 - 0s - loss: 0.6985 - acc: 0.8095
Epoch 178/500
 - 0s - loss: 0.6929 - acc: 0.8095
Epoch 179/500
 - 0s - loss: 0.6874 - acc: 0.8095
Epoch 180/500
 - 0s - loss: 0.6820 - acc: 0.8095
Epoch 181/500
 - 0s - loss: 0.6766 - acc: 0.8095
Epoch 182/500
 - 0s - loss: 0.6713 - acc: 0.8095
Epoch 183/500
 - 0s - loss: 0.6661 - acc: 0.8095
```

```
Epoch 184/500
 - 0s - loss: 0.6612 - acc: 0.8095
Epoch 185/500
 - 0s - loss: 0.6563 - acc: 0.8095
Epoch 186/500
 - 0s - loss: 0.6515 - acc: 0.8095
Epoch 187/500
 - 0s - loss: 0.6464 - acc: 0.8095
Epoch 188/500
 - 0s - loss: 0.6413 - acc: 0.8095
Epoch 189/500
 - 0s - loss: 0.6366 - acc: 0.8095
Epoch 190/500
 - 0s - loss: 0.6320 - acc: 0.8095
Epoch 191/500
 - 0s - loss: 0.6272 - acc: 0.8095
Epoch 192/500
 - 0s - loss: 0.6225 - acc: 0.8095
Epoch 193/500
 - 0s - loss: 0.6180 - acc: 0.8095
Epoch 194/500
 - 0s - loss: 0.6135 - acc: 0.8095
Epoch 195/500
 - 0s - loss: 0.6089 - acc: 0.8095
Epoch 196/500
 - 0s - loss: 0.6043 - acc: 0.8095
Epoch 197/500
 - 0s - loss: 0.6000 - acc: 0.8095
Epoch 198/500
 - 0s - loss: 0.5955 - acc: 0.8095
Epoch 199/500
 - 0s - loss: 0.5910 - acc: 0.8095
Epoch 200/500
 - 0s - loss: 0.5866 - acc: 0.8095
Epoch 201/500
 - 0s - loss: 0.5822 - acc: 0.8095
Epoch 202/500
 - 0s - loss: 0.5777 - acc: 0.8095
Epoch 203/500
 - 0s - loss: 0.5733 - acc: 0.8095
Epoch 204/500
 - 0s - loss: 0.5689 - acc: 0.8095
Epoch 205/500
 - 0s - loss: 0.5645 - acc: 0.8095
Epoch 206/500
 - 0s - loss: 0.5602 - acc: 0.8095
Epoch 207/500
 - 0s - loss: 0.5561 - acc: 0.8095
Epoch 208/500
 - 0s - loss: 0.5518 - acc: 0.8095
Epoch 209/500
 - 0s - loss: 0.5476 - acc: 0.8095
Epoch 210/500
 - 0s - loss: 0.5435 - acc: 0.8095
Epoch 211/500
 - 0s - loss: 0.5395 - acc: 0.8095
Epoch 212/500
 - 0s - loss: 0.5353 - acc: 0.8095
Epoch 213/500
 - 0s - loss: 0.5313 - acc: 0.8095
Epoch 214/500
```

```
 - 0s - loss: 0.5273 - acc: 0.8095
Epoch 215/500
 - 0s - loss: 0.5233 - acc: 0.8095
Epoch 216/500
 - 0s - loss: 0.5193 - acc: 0.8095
Epoch 217/500
 - 0s - loss: 0.5153 - acc: 0.8095
Epoch 218/500
 - 0s - loss: 0.5115 - acc: 0.8095
Epoch 219/500
 - 0s - loss: 0.5078 - acc: 0.8095
Epoch 220/500
 - 0s - loss: 0.5039 - acc: 0.8095
Epoch 221/500
 - 0s - loss: 0.5000 - acc: 0.8095
Epoch 222/500
 - 0s - loss: 0.4961 - acc: 0.8095
Epoch 223/500
 - 0s - loss: 0.4924 - acc: 0.8095
Epoch 224/500
 - 0s - loss: 0.4886 - acc: 0.8095
Epoch 225/500
 - 0s - loss: 0.4848 - acc: 0.8095
Epoch 226/500
 - 0s - loss: 0.4811 - acc: 0.8571
Epoch 227/500
 - 0s - loss: 0.4773 - acc: 0.8571
Epoch 228/500
 - 0s - loss: 0.4736 - acc: 0.8571
Epoch 229/500
 - 0s - loss: 0.4699 - acc: 0.8571
Epoch 230/500
 - 0s - loss: 0.4663 - acc: 0.8571
Epoch 231/500
 - 0s - loss: 0.4627 - acc: 0.8571
Epoch 232/500
 - 0s - loss: 0.4591 - acc: 0.8571
Epoch 233/500
 - 0s - loss: 0.4556 - acc: 0.8571
Epoch 234/500
 - 0s - loss: 0.4520 - acc: 0.9048
Epoch 235/500
 - 0s - loss: 0.4485 - acc: 0.9048
Epoch 236/500
 - 0s - loss: 0.4450 - acc: 0.9048
Epoch 237/500
 - 0s - loss: 0.4416 - acc: 0.9048
Epoch 238/500
 - 0s - loss: 0.4381 - acc: 0.9048
Epoch 239/500
 - 0s - loss: 0.4348 - acc: 0.9048
Epoch 240/500
 - 0s - loss: 0.4313 - acc: 0.9048
Epoch 241/500
 - 0s - loss: 0.4279 - acc: 0.9048
Epoch 242/500
 - 0s - loss: 0.4246 - acc: 0.9048
Epoch 243/500
 - 0s - loss: 0.4213 - acc: 0.9048
Epoch 244/500
 - 0s - loss: 0.4180 - acc: 0.9048
```

```
Epoch 245/500
 - 0s - loss: 0.4147 - acc: 0.9048
Epoch 246/500
 - 0s - loss: 0.4114 - acc: 0.9048
Epoch 247/500
 - 0s - loss: 0.4081 - acc: 0.9048
Epoch 248/500
 - 0s - loss: 0.4047 - acc: 0.9048
Epoch 249/500
 - 0s - loss: 0.4014 - acc: 0.9048
Epoch 250/500
 - 0s - loss: 0.3983 - acc: 0.9048
Epoch 251/500
 - 0s - loss: 0.3951 - acc: 0.9048
Epoch 252/500
 - 0s - loss: 0.3919 - acc: 0.9048
Epoch 253/500
 - 0s - loss: 0.3888 - acc: 0.9048
Epoch 254/500
 - 0s - loss: 0.3858 - acc: 0.9048
Epoch 255/500
 - 0s - loss: 0.3827 - acc: 0.9048
Epoch 256/500
 - 0s - loss: 0.3796 - acc: 0.9048
Epoch 257/500
 - 0s - loss: 0.3767 - acc: 0.9048
Epoch 258/500
 - 0s - loss: 0.3737 - acc: 0.9048
Epoch 259/500
 - 0s - loss: 0.3707 - acc: 0.9048
Epoch 260/500
 - 0s - loss: 0.3678 - acc: 0.9048
Epoch 261/500
 - 0s - loss: 0.3649 - acc: 0.9048
Epoch 262/500
 - 0s - loss: 0.3620 - acc: 0.9048
Epoch 263/500
 - 0s - loss: 0.3591 - acc: 0.9048
Epoch 264/500
 - 0s - loss: 0.3562 - acc: 0.9048
Epoch 265/500
 - 0s - loss: 0.3534 - acc: 0.9048
Epoch 266/500
 - 0s - loss: 0.3506 - acc: 0.9524
Epoch 267/500
 - 0s - loss: 0.3479 - acc: 0.9524
Epoch 268/500
 - 0s - loss: 0.3451 - acc: 0.9524
Epoch 269/500
 - 0s - loss: 0.3423 - acc: 0.9524
Epoch 270/500
 - 0s - loss: 0.3397 - acc: 0.9524
Epoch 271/500
 - 0s - loss: 0.3370 - acc: 0.9524
Epoch 272/500
 - 0s - loss: 0.3343 - acc: 0.9524
Epoch 273/500
 - 0s - loss: 0.3316 - acc: 0.9524
Epoch 274/500
 - 0s - loss: 0.3289 - acc: 0.9524
Epoch 275/500
```

```
 - 0s - loss: 0.3263 - acc: 0.9524
Epoch 276/500
 - 0s - loss: 0.3238 - acc: 0.9524
Epoch 277/500
 - 0s - loss: 0.3212 - acc: 0.9524
Epoch 278/500
 - 0s - loss: 0.3186 - acc: 0.9524
Epoch 279/500
 - 0s - loss: 0.3160 - acc: 0.9524
Epoch 280/500
 - 0s - loss: 0.3135 - acc: 0.9524
Epoch 281/500
 - 0s - loss: 0.3111 - acc: 0.9524
Epoch 282/500
 - 0s - loss: 0.3086 - acc: 0.9524
Epoch 283/500
 - 0s - loss: 0.3061 - acc: 0.9524
Epoch 284/500
 - 0s - loss: 0.3037 - acc: 0.9524
Epoch 285/500
 - 0s - loss: 0.3013 - acc: 0.9524
Epoch 286/500
 - 0s - loss: 0.2989 - acc: 0.9524
Epoch 287/500
 - 0s - loss: 0.2966 - acc: 0.9524
Epoch 288/500
 - 0s - loss: 0.2943 - acc: 0.9524
Epoch 289/500
 - 0s - loss: 0.2918 - acc: 0.9524
Epoch 290/500
 - 0s - loss: 0.2895 - acc: 0.9524
Epoch 291/500
 - 0s - loss: 0.2873 - acc: 0.9524
Epoch 292/500
 - 0s - loss: 0.2850 - acc: 0.9524
Epoch 293/500
 - 0s - loss: 0.2827 - acc: 0.9524
Epoch 294/500
 - 0s - loss: 0.2804 - acc: 0.9524
Epoch 295/500
 - 0s - loss: 0.2782 - acc: 0.9524
Epoch 296/500
 - 0s - loss: 0.2761 - acc: 0.9524
Epoch 297/500
 - 0s - loss: 0.2739 - acc: 0.9524
Epoch 298/500
 - 0s - loss: 0.2717 - acc: 0.9524
Epoch 299/500
 - 0s - loss: 0.2696 - acc: 0.9524
Epoch 300/500
 - 0s - loss: 0.2674 - acc: 0.9524
Epoch 301/500
 - 0s - loss: 0.2653 - acc: 0.9524
Epoch 302/500
 - 0s - loss: 0.2633 - acc: 0.9524
Epoch 303/500
 - 0s - loss: 0.2612 - acc: 0.9524
Epoch 304/500
 - 0s - loss: 0.2591 - acc: 0.9524
Epoch 305/500
 - 0s - loss: 0.2571 - acc: 0.9524
```

```
Epoch 306/500
 - 0s - loss: 0.2551 - acc: 0.9524
Epoch 307/500
 - 0s - loss: 0.2531 - acc: 0.9524
Epoch 308/500
 - 0s - loss: 0.2511 - acc: 0.9524
Epoch 309/500
 - 0s - loss: 0.2492 - acc: 0.9524
Epoch 310/500
 - 0s - loss: 0.2472 - acc: 0.9524
Epoch 311/500
 - 0s - loss: 0.2453 - acc: 0.9524
Epoch 312/500
 - 0s - loss: 0.2434 - acc: 0.9524
Epoch 313/500
 - 0s - loss: 0.2415 - acc: 0.9524
Epoch 314/500
 - 0s - loss: 0.2397 - acc: 0.9524
Epoch 315/500
 - 0s - loss: 0.2378 - acc: 0.9524
Epoch 316/500
 - 0s - loss: 0.2360 - acc: 0.9524
Epoch 317/500
 - 0s - loss: 0.2342 - acc: 0.9524
Epoch 318/500
 - 0s - loss: 0.2324 - acc: 0.9524
Epoch 319/500
 - 0s - loss: 0.2307 - acc: 0.9524
Epoch 320/500
 - 0s - loss: 0.2288 - acc: 0.9524
Epoch 321/500
 - 0s - loss: 0.2271 - acc: 0.9524
Epoch 322/500
 - 0s - loss: 0.2254 - acc: 0.9524
Epoch 323/500
 - 0s - loss: 0.2236 - acc: 0.9524
Epoch 324/500
 - 0s - loss: 0.2220 - acc: 0.9524
Epoch 325/500
 - 0s - loss: 0.2203 - acc: 0.9524
Epoch 326/500
 - 0s - loss: 0.2186 - acc: 0.9524
Epoch 327/500
 - 0s - loss: 0.2170 - acc: 0.9524
Epoch 328/500
 - 0s - loss: 0.2154 - acc: 0.9524
Epoch 329/500
 - 0s - loss: 0.2138 - acc: 0.9524
Epoch 330/500
 - 0s - loss: 0.2121 - acc: 0.9524
Epoch 331/500
 - 0s - loss: 0.2105 - acc: 0.9524
Epoch 332/500
 - 0s - loss: 0.2090 - acc: 0.9524
Epoch 333/500
 - 0s - loss: 0.2074 - acc: 0.9524
Epoch 334/500
 - 0s - loss: 0.2058 - acc: 0.9524
Epoch 335/500
 - 0s - loss: 0.2043 - acc: 0.9524
Epoch 336/500
```

```
 - 0s - loss: 0.2028 - acc: 0.9524
Epoch 337/500
 - 0s - loss: 0.2014 - acc: 0.9524
Epoch 338/500
 - 0s - loss: 0.1999 - acc: 0.9524
Epoch 339/500
 - 0s - loss: 0.1984 - acc: 0.9524
Epoch 340/500
 - 0s - loss: 0.1970 - acc: 0.9524
Epoch 341/500
 - 0s - loss: 0.1956 - acc: 0.9524
Epoch 342/500
 - 0s - loss: 0.1942 - acc: 0.9524
Epoch 343/500
 - 0s - loss: 0.1928 - acc: 0.9524
Epoch 344/500
 - 0s - loss: 0.1914 - acc: 0.9524
Epoch 345/500
 - 0s - loss: 0.1900 - acc: 0.9524
Epoch 346/500
 - 0s - loss: 0.1887 - acc: 0.9524
Epoch 347/500
 - 0s - loss: 0.1874 - acc: 0.9524
Epoch 348/500
 - 0s - loss: 0.1861 - acc: 0.9524
Epoch 349/500
 - 0s - loss: 0.1848 - acc: 0.9524
Epoch 350/500
 - 0s - loss: 0.1835 - acc: 0.9524
Epoch 351/500
 - 0s - loss: 0.1822 - acc: 0.9524
Epoch 352/500
 - 0s - loss: 0.1809 - acc: 0.9524
Epoch 353/500
 - 0s - loss: 0.1797 - acc: 0.9524
Epoch 354/500
 - 0s - loss: 0.1785 - acc: 0.9524
Epoch 355/500
 - 0s - loss: 0.1773 - acc: 0.9524
Epoch 356/500
 - 0s - loss: 0.1761 - acc: 0.9524
Epoch 357/500
 - 0s - loss: 0.1749 - acc: 0.9524
Epoch 358/500
 - 0s - loss: 0.1738 - acc: 0.9524
Epoch 359/500
 - 0s - loss: 0.1726 - acc: 0.9524
Epoch 360/500
 - 0s - loss: 0.1715 - acc: 0.9524
Epoch 361/500
 - 0s - loss: 0.1703 - acc: 0.9524
Epoch 362/500
 - 0s - loss: 0.1692 - acc: 0.9524
Epoch 363/500
 - 0s - loss: 0.1681 - acc: 0.9524
Epoch 364/500
 - 0s - loss: 0.1670 - acc: 0.9524
Epoch 365/500
 - 0s - loss: 0.1660 - acc: 0.9524
Epoch 366/500
 - 0s - loss: 0.1649 - acc: 0.9524
```

```
Epoch 367/500
 - 0s - loss: 0.1639 - acc: 0.9524
Epoch 368/500
 - 0s - loss: 0.1628 - acc: 0.9524
Epoch 369/500
 - 0s - loss: 0.1618 - acc: 0.9524
Epoch 370/500
 - 0s - loss: 0.1608 - acc: 0.9524
Epoch 371/500
 - 0s - loss: 0.1598 - acc: 0.9524
Epoch 372/500
 - 0s - loss: 0.1589 - acc: 0.9524
Epoch 373/500
 - 0s - loss: 0.1579 - acc: 0.9524
Epoch 374/500
 - 0s - loss: 0.1569 - acc: 0.9524
Epoch 375/500
 - 0s - loss: 0.1560 - acc: 0.9524
Epoch 376/500
 - 0s - loss: 0.1551 - acc: 0.9524
Epoch 377/500
 - 0s - loss: 0.1541 - acc: 0.9524
Epoch 378/500
 - 0s - loss: 0.1532 - acc: 0.9524
Epoch 379/500
 - 0s - loss: 0.1523 - acc: 0.9524
Epoch 380/500
 - 0s - loss: 0.1514 - acc: 0.9524
Epoch 381/500
 - 0s - loss: 0.1506 - acc: 0.9524
Epoch 382/500
 - 0s - loss: 0.1497 - acc: 0.9524
Epoch 383/500
 - 0s - loss: 0.1489 - acc: 0.9524
Epoch 384/500
 - 0s - loss: 0.1480 - acc: 0.9524
Epoch 385/500
 - 0s - loss: 0.1472 - acc: 0.9524
Epoch 386/500
 - 0s - loss: 0.1464 - acc: 0.9524
Epoch 387/500
 - 0s - loss: 0.1455 - acc: 0.9524
Epoch 388/500
 - 0s - loss: 0.1448 - acc: 0.9524
Epoch 389/500
 - 0s - loss: 0.1440 - acc: 0.9524
Epoch 390/500
 - 0s - loss: 0.1432 - acc: 0.9524
Epoch 391/500
 - 0s - loss: 0.1424 - acc: 0.9524
Epoch 392/500
 - 0s - loss: 0.1417 - acc: 0.9524
Epoch 393/500
 - 0s - loss: 0.1409 - acc: 0.9524
Epoch 394/500
 - 0s - loss: 0.1402 - acc: 0.9524
Epoch 395/500
 - 0s - loss: 0.1395 - acc: 0.9524
Epoch 396/500
 - 0s - loss: 0.1388 - acc: 0.9524
Epoch 397/500
```

```
 - 0s - loss: 0.1380 - acc: 0.9524
Epoch 398/500
 - 0s - loss: 0.1373 - acc: 0.9524
Epoch 399/500
 - 0s - loss: 0.1366 - acc: 0.9524
Epoch 400/500
 - 0s - loss: 0.1360 - acc: 0.9524
Epoch 401/500
 - 0s - loss: 0.1353 - acc: 0.9524
Epoch 402/500
 - 0s - loss: 0.1346 - acc: 0.9524
Epoch 403/500
 - 0s - loss: 0.1340 - acc: 0.9524
Epoch 404/500
 - 0s - loss: 0.1333 - acc: 0.9524
Epoch 405/500
 - 0s - loss: 0.1327 - acc: 0.9524
Epoch 406/500
 - 0s - loss: 0.1321 - acc: 0.9524
Epoch 407/500
 - 0s - loss: 0.1314 - acc: 0.9524
Epoch 408/500
 - 0s - loss: 0.1308 - acc: 0.9524
Epoch 409/500
 - 0s - loss: 0.1302 - acc: 0.9524
Epoch 410/500
 - 0s - loss: 0.1296 - acc: 0.9524
Epoch 411/500
 - 0s - loss: 0.1290 - acc: 0.9524
Epoch 412/500
 - 0s - loss: 0.1284 - acc: 0.9524
Epoch 413/500
 - 0s - loss: 0.1279 - acc: 0.9524
Epoch 414/500
 - 0s - loss: 0.1273 - acc: 0.9524
Epoch 415/500
 - 0s - loss: 0.1267 - acc: 0.9524
Epoch 416/500
 - 0s - loss: 0.1262 - acc: 0.9524
Epoch 417/500
 - 0s - loss: 0.1256 - acc: 0.9524
Epoch 418/500
 - 0s - loss: 0.1251 - acc: 0.9524
Epoch 419/500
 - 0s - loss: 0.1246 - acc: 0.9524
Epoch 420/500
 - 0s - loss: 0.1240 - acc: 0.9524
Epoch 421/500
 - 0s - loss: 0.1235 - acc: 0.9524
Epoch 422/500
 - 0s - loss: 0.1230 - acc: 0.9524
Epoch 423/500
 - 0s - loss: 0.1225 - acc: 0.9524
Epoch 424/500
 - 0s - loss: 0.1220 - acc: 0.9524
Epoch 425/500
 - 0s - loss: 0.1215 - acc: 0.9524
Epoch 426/500
 - 0s - loss: 0.1210 - acc: 0.9524
Epoch 427/500
 - 0s - loss: 0.1205 - acc: 0.9524
```

```
Epoch 428/500
 - 0s - loss: 0.1201 - acc: 0.9524
Epoch 429/500
 - 0s - loss: 0.1196 - acc: 0.9524
Epoch 430/500
 - 0s - loss: 0.1191 - acc: 0.9524
Epoch 431/500
 - 0s - loss: 0.1187 - acc: 0.9524
Epoch 432/500
 - 0s - loss: 0.1182 - acc: 0.9524
Epoch 433/500
 - 0s - loss: 0.1178 - acc: 0.9524
Epoch 434/500
 - 0s - loss: 0.1173 - acc: 0.9524
Epoch 435/500
 - 0s - loss: 0.1169 - acc: 0.9524
Epoch 436/500
 - 0s - loss: 0.1165 - acc: 0.9524
Epoch 437/500
 - 0s - loss: 0.1161 - acc: 0.9524
Epoch 438/500
 - 0s - loss: 0.1156 - acc: 0.9524
Epoch 439/500
 - 0s - loss: 0.1152 - acc: 0.9524
Epoch 440/500
 - 0s - loss: 0.1148 - acc: 0.9524
Epoch 441/500
 - 0s - loss: 0.1144 - acc: 0.9524
Epoch 442/500
 - 0s - loss: 0.1140 - acc: 0.9524
Epoch 443/500
 - 0s - loss: 0.1137 - acc: 0.9524
Epoch 444/500
 - 0s - loss: 0.1132 - acc: 0.9524
Epoch 445/500
 - 0s - loss: 0.1129 - acc: 0.9524
Epoch 446/500
 - 0s - loss: 0.1125 - acc: 0.9524
Epoch 447/500
 - 0s - loss: 0.1121 - acc: 0.9524
Epoch 448/500
 - 0s - loss: 0.1117 - acc: 0.9524
Epoch 449/500
 - 0s - loss: 0.1114 - acc: 0.9524
Epoch 450/500
 - 0s - loss: 0.1110 - acc: 0.9524
Epoch 451/500
 - 0s - loss: 0.1107 - acc: 0.9524
Epoch 452/500
 - 0s - loss: 0.1103 - acc: 0.9524
Epoch 453/500
 - 0s - loss: 0.1100 - acc: 0.9524
Epoch 454/500
 - 0s - loss: 0.1096 - acc: 0.9524
Epoch 455/500
 - 0s - loss: 0.1093 - acc: 0.9524
Epoch 456/500
 - 0s - loss: 0.1090 - acc: 0.9524
Epoch 457/500
 - 0s - loss: 0.1086 - acc: 0.9524
Epoch 458/500
```

```
 - 0s - loss: 0.1083 - acc: 0.9524
Epoch 459/500
 - 0s - loss: 0.1080 - acc: 0.9524
Epoch 460/500
 - 0s - loss: 0.1077 - acc: 0.9524
Epoch 461/500
 - 0s - loss: 0.1073 - acc: 0.9524
Epoch 462/500
 - 0s - loss: 0.1070 - acc: 0.9524
Epoch 463/500
 - 0s - loss: 0.1067 - acc: 0.9524
Epoch 464/500
 - 0s - loss: 0.1064 - acc: 0.9524
Epoch 465/500
 - 0s - loss: 0.1061 - acc: 0.9524
Epoch 466/500
 - 0s - loss: 0.1058 - acc: 0.9524
Epoch 467/500
 - 0s - loss: 0.1055 - acc: 0.9524
Epoch 468/500
 - 0s - loss: 0.1052 - acc: 0.9524
Epoch 469/500
 - 0s - loss: 0.1049 - acc: 0.9524
Epoch 470/500
 - 0s - loss: 0.1046 - acc: 0.9524
Epoch 471/500
 - 0s - loss: 0.1043 - acc: 0.9524
Epoch 472/500
 - 0s - loss: 0.1041 - acc: 0.9524
Epoch 473/500
 - 0s - loss: 0.1038 - acc: 0.9524
Epoch 474/500
 - 0s - loss: 0.1035 - acc: 0.9524
Epoch 475/500
 - 0s - loss: 0.1032 - acc: 0.9524
Epoch 476/500
 - 0s - loss: 0.1030 - acc: 0.9524
Epoch 477/500
 - 0s - loss: 0.1027 - acc: 0.9524
Epoch 478/500
 - 0s - loss: 0.1024 - acc: 0.9524
Epoch 479/500
 - 0s - loss: 0.1021 - acc: 0.9524
Epoch 480/500
 - 0s - loss: 0.1019 - acc: 0.9524
Epoch 481/500
 - 0s - loss: 0.1016 - acc: 0.9524
Epoch 482/500
 - 0s - loss: 0.1014 - acc: 0.9524
Epoch 483/500
 - 0s - loss: 0.1011 - acc: 0.9524
Epoch 484/500
 - 0s - loss: 0.1009 - acc: 0.9524
Epoch 485/500
 - 0s - loss: 0.1006 - acc: 0.9524
Epoch 486/500
 - 0s - loss: 0.1004 - acc: 0.9524
Epoch 487/500
 - 0s - loss: 0.1001 - acc: 0.9524
Epoch 488/500
 - 0s - loss: 0.0999 - acc: 0.9524
```

```
Epoch 489/500
 - 0s - loss: 0.0997 - acc: 0.9524
Epoch 490/500
 - 0s - loss: 0.0994 - acc: 0.9524
Epoch 491/500
 - 0s - loss: 0.0992 - acc: 0.9524
Epoch 492/500
 - 0s - loss: 0.0990 - acc: 0.9524
Epoch 493/500
 - 0s - loss: 0.0987 - acc: 0.9524
Epoch 494/500
 - 0s - loss: 0.0985 - acc: 0.9524
Epoch 495/500
 - 0s - loss: 0.0983 - acc: 0.9524
Epoch 496/500
 - 0s - loss: 0.0981 - acc: 0.9524
Epoch 497/500
 - 0s - loss: 0.0979 - acc: 0.9524
Epoch 498/500
 - 0s - loss: 0.0976 - acc: 0.9524
Epoch 499/500
 - 0s - loss: 0.0974 - acc: 0.9524
Epoch 500/500
 - 0s - loss: 0.0972 - acc: 0.9524
```

Out[24]:

```
<keras.callbacks.History at 0x2e6dfc36390>
```

We can use the model to generate new sequences as before. The `generate_seq()` function can be updated to build up an input sequence by adding predictions to the list of input words each iteration.

In [25]:

```
# generate a sequence from a language model
def generate_seq(model, tokenizer, max_length, seed_text, n_words):
    in_text = seed_text
    # generate a fixed number of words
    for _ in range(n_words):
        # encode the text as integer
        encoded = tokenizer.texts_to_sequences([in_text])[0]
        # pre-pad sequences to a fixed length
        encoded = pad_sequences([encoded], maxlen=max_length, padding='pre')
        # predict probabilities for each word
        yhat = model.predict_classes(encoded, verbose=0)
        # map predicted word index to word
        out_word = ''
        for word, index in tokenizer.word_index.items():
            if index == yhat:
                out_word = word
                break
        # append to input
        in_text += ' ' + out_word
    return in_text
```

In [26]:

```
# evaluate model
print(generate_seq(model, tokenizer, max_length-1, 'Jack', 4))
```

```
Jack fell down and broke
```

In [27]:

```
print(generate_seq(model, tokenizer, max_length-1, 'Jill', 4))
```

```
Jill jill came tumbling after
```

Model 3: Two-Words-In, One-Word-Out Sequence

In [28]:

```
from numpy import array
from keras.preprocessing.text import Tokenizer
from keras.utils import to_categorical
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Embedding
```

In [29]:

```
# source text
data = """ Jack and Jill went up the hill\n
        To fetch a pail of water\n
        Jack fell down and broke his crown\n
        And Jill came tumbling after\n """
```

In [30]:

```
# integer encode sequences of words
tokenizer = Tokenizer()
tokenizer.fit_on_texts([data])
encoded = tokenizer.texts_to_sequences([data])[0]
```

In [31]:

```
# retrieve vocabulary size
vocab_size = len(tokenizer.word_index) + 1
print('Vocabulary Size: %d' % vocab_size)
```

```
Vocabulary Size: 22
```

We will use 3 words as input to predict one word as output. The preparation of the sequences is much like the first example, except with different offsets in the source sequence arrays

In [32]:

```
# encode 2 words -> 1 word
sequences = list()
for i in range(2, len(encoded)):
    sequence = encoded[i-2:i+1]
    sequences.append(sequence)
print('Total Sequences: %d' % len(sequences))
```

```
Total Sequences: 23
```

In [33]:

```
# pad sequences
max_length = max([len(seq) for seq in sequences])
sequences = pad_sequences(sequences, maxlen=max_length, padding='pre')
print('Max Sequence Length: %d' % max_length)
```

```
Max Sequence Length: 3
```

In [34]:

```
# split into input and output elements
sequences = array(sequences)
X, y = sequences[:,:-1],sequences[:,-1]
y = to_categorical(y, num_classes=vocab_size)
```

In [35]:

```
# define model
model = Sequential()
model.add(Embedding(vocab_size, 10, input_length=max_length-1))
model.add(LSTM(50))
model.add(Dense(vocab_size, activation='softmax'))
print(model.summary())
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_3 (Embedding)      (None, 2, 10)             220
_________________________________________________________________
lstm_3 (LSTM)                (None, 50)                12200
_________________________________________________________________
dense_3 (Dense)              (None, 22)                1122
=================================================================
Total params: 13,542
Trainable params: 13,542
Non-trainable params: 0
_________________________________________________________________
None
```

In [36]:

```
# compile network
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

In [37]:

```
# fit network
model.fit(X, y, epochs=500, verbose=2)
```

```
Epoch 1/500
 - 1s - loss: 3.0902 - acc: 0.0870
Epoch 2/500
 - 0s - loss: 3.0893 - acc: 0.0435
Epoch 3/500
 - 0s - loss: 3.0885 - acc: 0.1304
Epoch 4/500
 - 0s - loss: 3.0876 - acc: 0.0870
Epoch 5/500
 - 0s - loss: 3.0867 - acc: 0.0870
Epoch 6/500
 - 0s - loss: 3.0858 - acc: 0.0870
Epoch 7/500
 - 0s - loss: 3.0849 - acc: 0.0870
Epoch 8/500
 - 0s - loss: 3.0840 - acc: 0.0870
Epoch 9/500
 - 0s - loss: 3.0831 - acc: 0.0870
Epoch 10/500
 - 0s - loss: 3.0821 - acc: 0.0870
Epoch 11/500
 - 0s - loss: 3.0811 - acc: 0.0870
Epoch 12/500
 - 0s - loss: 3.0801 - acc: 0.0870
Epoch 13/500
 - 0s - loss: 3.0790 - acc: 0.0870
Epoch 14/500
 - 0s - loss: 3.0780 - acc: 0.0870
Epoch 15/500
 - 0s - loss: 3.0769 - acc: 0.0870
Epoch 16/500
 - 0s - loss: 3.0757 - acc: 0.0870
Epoch 17/500
 - 0s - loss: 3.0746 - acc: 0.0870
Epoch 18/500
 - 0s - loss: 3.0734 - acc: 0.0870
Epoch 19/500
 - 0s - loss: 3.0721 - acc: 0.0870
Epoch 20/500
 - 0s - loss: 3.0708 - acc: 0.0870
Epoch 21/500
 - 0s - loss: 3.0695 - acc: 0.0870
Epoch 22/500
 - 0s - loss: 3.0682 - acc: 0.0870
Epoch 23/500
 - 0s - loss: 3.0668 - acc: 0.0870
Epoch 24/500
 - 0s - loss: 3.0653 - acc: 0.0870
Epoch 25/500
 - 0s - loss: 3.0638 - acc: 0.0870
Epoch 26/500
 - 0s - loss: 3.0622 - acc: 0.0870
Epoch 27/500
 - 0s - loss: 3.0606 - acc: 0.0870
Epoch 28/500
 - 0s - loss: 3.0589 - acc: 0.0870
Epoch 29/500
 - 0s - loss: 3.0571 - acc: 0.0870
Epoch 30/500
 - 0s - loss: 3.0553 - acc: 0.0870
Epoch 31/500
```

```
 - 0s - loss: 3.0534 - acc: 0.0870
Epoch 32/500
 - 0s - loss: 3.0515 - acc: 0.0870
Epoch 33/500
 - 0s - loss: 3.0494 - acc: 0.0870
Epoch 34/500
 - 0s - loss: 3.0473 - acc: 0.0870
Epoch 35/500
 - 0s - loss: 3.0451 - acc: 0.0870
Epoch 36/500
 - 0s - loss: 3.0429 - acc: 0.0870
Epoch 37/500
 - 0s - loss: 3.0405 - acc: 0.0870
Epoch 38/500
 - 0s - loss: 3.0380 - acc: 0.0870
Epoch 39/500
 - 0s - loss: 3.0355 - acc: 0.0870
Epoch 40/500
 - 0s - loss: 3.0328 - acc: 0.0870
Epoch 41/500
 - 0s - loss: 3.0300 - acc: 0.0870
Epoch 42/500
 - 0s - loss: 3.0272 - acc: 0.0870
Epoch 43/500
 - 0s - loss: 3.0242 - acc: 0.0870
Epoch 44/500
 - 0s - loss: 3.0211 - acc: 0.0870
Epoch 45/500
 - 0s - loss: 3.0179 - acc: 0.0870
Epoch 46/500
 - 0s - loss: 3.0145 - acc: 0.0870
Epoch 47/500
 - 0s - loss: 3.0110 - acc: 0.0870
Epoch 48/500
 - 0s - loss: 3.0074 - acc: 0.0870
Epoch 49/500
 - 0s - loss: 3.0036 - acc: 0.0870
Epoch 50/500
 - 0s - loss: 2.9997 - acc: 0.0870
Epoch 51/500
 - 0s - loss: 2.9956 - acc: 0.0870
Epoch 52/500
 - 0s - loss: 2.9914 - acc: 0.0870
Epoch 53/500
 - 0s - loss: 2.9869 - acc: 0.0870
Epoch 54/500
 - 0s - loss: 2.9823 - acc: 0.0870
Epoch 55/500
 - 0s - loss: 2.9775 - acc: 0.0870
Epoch 56/500
 - 0s - loss: 2.9726 - acc: 0.0870
Epoch 57/500
 - 0s - loss: 2.9674 - acc: 0.0870
Epoch 58/500
 - 0s - loss: 2.9620 - acc: 0.0870
Epoch 59/500
 - 0s - loss: 2.9564 - acc: 0.0870
Epoch 60/500
 - 0s - loss: 2.9506 - acc: 0.0870
Epoch 61/500
 - 0s - loss: 2.9445 - acc: 0.0870
```

```
Epoch 62/500
 - 0s - loss: 2.9383 - acc: 0.0870
Epoch 63/500
 - 0s - loss: 2.9317 - acc: 0.0870
Epoch 64/500
 - 0s - loss: 2.9250 - acc: 0.0870
Epoch 65/500
 - 0s - loss: 2.9179 - acc: 0.0870
Epoch 66/500
 - 0s - loss: 2.9105 - acc: 0.0870
Epoch 67/500
 - 0s - loss: 2.9030 - acc: 0.0870
Epoch 68/500
 - 0s - loss: 2.8951 - acc: 0.0870
Epoch 69/500
 - 0s - loss: 2.8870 - acc: 0.0870
Epoch 70/500
 - 0s - loss: 2.8785 - acc: 0.0870
Epoch 71/500
 - 0s - loss: 2.8698 - acc: 0.0870
Epoch 72/500
 - 0s - loss: 2.8608 - acc: 0.0870
Epoch 73/500
 - 0s - loss: 2.8513 - acc: 0.0870
Epoch 74/500
 - 0s - loss: 2.8416 - acc: 0.0870
Epoch 75/500
 - 0s - loss: 2.8316 - acc: 0.1304
Epoch 76/500
 - 0s - loss: 2.8212 - acc: 0.1304
Epoch 77/500
 - 0s - loss: 2.8106 - acc: 0.1304
Epoch 78/500
 - 0s - loss: 2.7995 - acc: 0.1304
Epoch 79/500
 - 0s - loss: 2.7881 - acc: 0.1304
Epoch 80/500
 - 0s - loss: 2.7763 - acc: 0.1304
Epoch 81/500
 - 0s - loss: 2.7642 - acc: 0.1304
Epoch 82/500
 - 0s - loss: 2.7518 - acc: 0.1304
Epoch 83/500
 - 0s - loss: 2.7391 - acc: 0.1304
Epoch 84/500
 - 0s - loss: 2.7260 - acc: 0.1304
Epoch 85/500
 - 0s - loss: 2.7125 - acc: 0.1304
Epoch 86/500
 - 0s - loss: 2.6987 - acc: 0.1739
Epoch 87/500
 - 0s - loss: 2.6846 - acc: 0.1739
Epoch 88/500
 - 0s - loss: 2.6701 - acc: 0.1739
Epoch 89/500
 - 0s - loss: 2.6552 - acc: 0.1739
Epoch 90/500
 - 0s - loss: 2.6401 - acc: 0.2174
Epoch 91/500
 - 0s - loss: 2.6246 - acc: 0.2174
Epoch 92/500
```

```
 - 0s - loss: 2.6088 - acc: 0.2174
Epoch 93/500
 - 0s - loss: 2.5927 - acc: 0.2174
Epoch 94/500
 - 0s - loss: 2.5763 - acc: 0.2609
Epoch 95/500
 - 0s - loss: 2.5596 - acc: 0.2609
Epoch 96/500
 - 0s - loss: 2.5427 - acc: 0.2609
Epoch 97/500
 - 0s - loss: 2.5254 - acc: 0.2609
Epoch 98/500
 - 0s - loss: 2.5079 - acc: 0.2609
Epoch 99/500
 - 0s - loss: 2.4902 - acc: 0.2609
Epoch 100/500
 - 0s - loss: 2.4721 - acc: 0.2609
Epoch 101/500
 - 0s - loss: 2.4539 - acc: 0.2609
Epoch 102/500
 - 0s - loss: 2.4355 - acc: 0.2609
Epoch 103/500
 - 0s - loss: 2.4168 - acc: 0.2609
Epoch 104/500
 - 0s - loss: 2.3980 - acc: 0.2609
Epoch 105/500
 - 0s - loss: 2.3790 - acc: 0.2609
Epoch 106/500
 - 0s - loss: 2.3599 - acc: 0.2609
Epoch 107/500
 - 0s - loss: 2.3405 - acc: 0.2609
Epoch 108/500
 - 0s - loss: 2.3211 - acc: 0.2609
Epoch 109/500
 - 0s - loss: 2.3015 - acc: 0.2609
Epoch 110/500
 - 0s - loss: 2.2818 - acc: 0.2609
Epoch 111/500
 - 0s - loss: 2.2619 - acc: 0.3043
Epoch 112/500
 - 0s - loss: 2.2419 - acc: 0.3043
Epoch 113/500
 - 0s - loss: 2.2218 - acc: 0.3043
Epoch 114/500
 - 0s - loss: 2.2016 - acc: 0.3913
Epoch 115/500
 - 0s - loss: 2.1813 - acc: 0.3913
Epoch 116/500
 - 0s - loss: 2.1609 - acc: 0.3913
Epoch 117/500
 - 0s - loss: 2.1404 - acc: 0.4783
Epoch 118/500
 - 0s - loss: 2.1198 - acc: 0.4783
Epoch 119/500
 - 0s - loss: 2.0990 - acc: 0.5217
Epoch 120/500
 - 0s - loss: 2.0782 - acc: 0.5217
Epoch 121/500
 - 0s - loss: 2.0572 - acc: 0.5217
Epoch 122/500
 - 0s - loss: 2.0362 - acc: 0.5217
```

```
Epoch 123/500
 - 0s - loss: 2.0150 - acc: 0.5652
Epoch 124/500
 - 0s - loss: 1.9938 - acc: 0.6087
Epoch 125/500
 - 0s - loss: 1.9725 - acc: 0.6087
Epoch 126/500
 - 0s - loss: 1.9510 - acc: 0.6087
Epoch 127/500
 - 0s - loss: 1.9295 - acc: 0.6087
Epoch 128/500
 - 0s - loss: 1.9079 - acc: 0.6087
Epoch 129/500
 - 0s - loss: 1.8862 - acc: 0.6087
Epoch 130/500
 - 0s - loss: 1.8643 - acc: 0.6522
Epoch 131/500
 - 0s - loss: 1.8424 - acc: 0.6522
Epoch 132/500
 - 0s - loss: 1.8203 - acc: 0.6522
Epoch 133/500
 - 0s - loss: 1.7982 - acc: 0.6522
Epoch 134/500
 - 0s - loss: 1.7760 - acc: 0.6522
Epoch 135/500
 - 0s - loss: 1.7538 - acc: 0.6522
Epoch 136/500
 - 0s - loss: 1.7315 - acc: 0.6957
Epoch 137/500
 - 0s - loss: 1.7092 - acc: 0.6957
Epoch 138/500
 - 0s - loss: 1.6869 - acc: 0.6957
Epoch 139/500
 - 0s - loss: 1.6645 - acc: 0.6957
Epoch 140/500
 - 0s - loss: 1.6420 - acc: 0.6957
Epoch 141/500
 - 0s - loss: 1.6196 - acc: 0.6957
Epoch 142/500
 - 0s - loss: 1.5972 - acc: 0.6957
Epoch 143/500
 - 0s - loss: 1.5747 - acc: 0.7391
Epoch 144/500
 - 0s - loss: 1.5523 - acc: 0.7391
Epoch 145/500
 - 0s - loss: 1.5299 - acc: 0.7391
Epoch 146/500
 - 0s - loss: 1.5076 - acc: 0.7391
Epoch 147/500
 - 0s - loss: 1.4853 - acc: 0.7391
Epoch 148/500
 - 0s - loss: 1.4631 - acc: 0.7391
Epoch 149/500
 - 0s - loss: 1.4409 - acc: 0.7391
Epoch 150/500
 - 0s - loss: 1.4188 - acc: 0.7391
Epoch 151/500
 - 0s - loss: 1.3968 - acc: 0.7826
Epoch 152/500
 - 0s - loss: 1.3748 - acc: 0.7826
Epoch 153/500
```

```
 - 0s - loss: 1.3529 - acc: 0.7826
Epoch 154/500
 - 0s - loss: 1.3312 - acc: 0.8696
Epoch 155/500
 - 0s - loss: 1.3095 - acc: 0.8696
Epoch 156/500
 - 0s - loss: 1.2880 - acc: 0.8696
Epoch 157/500
 - 0s - loss: 1.2666 - acc: 0.8696
Epoch 158/500
 - 0s - loss: 1.2453 - acc: 0.8696
Epoch 159/500
 - 0s - loss: 1.2242 - acc: 0.8696
Epoch 160/500
 - 0s - loss: 1.2031 - acc: 0.8696
Epoch 161/500
 - 0s - loss: 1.1823 - acc: 0.8696
Epoch 162/500
 - 0s - loss: 1.1616 - acc: 0.8696
Epoch 163/500
 - 0s - loss: 1.1411 - acc: 0.8696
Epoch 164/500
 - 0s - loss: 1.1208 - acc: 0.8696
Epoch 165/500
 - 0s - loss: 1.1006 - acc: 0.9130
Epoch 166/500
 - 0s - loss: 1.0806 - acc: 0.9130
Epoch 167/500
 - 0s - loss: 1.0608 - acc: 0.9130
Epoch 168/500
 - 0s - loss: 1.0412 - acc: 0.9130
Epoch 169/500
 - 0s - loss: 1.0217 - acc: 0.9130
Epoch 170/500
 - 0s - loss: 1.0025 - acc: 0.9130
Epoch 171/500
 - 0s - loss: 0.9835 - acc: 0.9130
Epoch 172/500
 - 0s - loss: 0.9647 - acc: 0.9130
Epoch 173/500
 - 0s - loss: 0.9461 - acc: 0.9130
Epoch 174/500
 - 0s - loss: 0.9278 - acc: 0.9130
Epoch 175/500
 - 0s - loss: 0.9097 - acc: 0.9130
Epoch 176/500
 - 0s - loss: 0.8918 - acc: 0.9565
Epoch 177/500
 - 0s - loss: 0.8741 - acc: 0.9565
Epoch 178/500
 - 0s - loss: 0.8568 - acc: 0.9565
Epoch 179/500
 - 0s - loss: 0.8398 - acc: 0.9565
Epoch 180/500
 - 0s - loss: 0.8231 - acc: 0.9565
Epoch 181/500
 - 0s - loss: 0.8066 - acc: 0.9565
Epoch 182/500
 - 0s - loss: 0.7904 - acc: 0.9565
Epoch 183/500
 - 0s - loss: 0.7744 - acc: 0.9565
```

```
Epoch 184/500
 - 0s - loss: 0.7588 - acc: 0.9565
Epoch 185/500
 - 0s - loss: 0.7435 - acc: 0.9565
Epoch 186/500
 - 0s - loss: 0.7285 - acc: 0.9565
Epoch 187/500
 - 0s - loss: 0.7137 - acc: 0.9565
Epoch 188/500
 - 0s - loss: 0.6993 - acc: 0.9565
Epoch 189/500
 - 0s - loss: 0.6852 - acc: 0.9565
Epoch 190/500
 - 0s - loss: 0.6712 - acc: 0.9565
Epoch 191/500
 - 0s - loss: 0.6575 - acc: 0.9565
Epoch 192/500
 - 0s - loss: 0.6440 - acc: 0.9565
Epoch 193/500
 - 0s - loss: 0.6307 - acc: 0.9565
Epoch 194/500
 - 0s - loss: 0.6177 - acc: 0.9565
Epoch 195/500
 - 0s - loss: 0.6050 - acc: 0.9565
Epoch 196/500
 - 0s - loss: 0.5925 - acc: 0.9565
Epoch 197/500
 - 0s - loss: 0.5802 - acc: 0.9565
Epoch 198/500
 - 0s - loss: 0.5681 - acc: 0.9565
Epoch 199/500
 - 0s - loss: 0.5562 - acc: 0.9565
Epoch 200/500
 - 0s - loss: 0.5446 - acc: 0.9565
Epoch 201/500
 - 0s - loss: 0.5331 - acc: 0.9565
Epoch 202/500
 - 0s - loss: 0.5219 - acc: 0.9565
Epoch 203/500
 - 0s - loss: 0.5108 - acc: 0.9565
Epoch 204/500
 - 0s - loss: 0.5000 - acc: 0.9565
Epoch 205/500
 - 0s - loss: 0.4894 - acc: 0.9565
Epoch 206/500
 - 0s - loss: 0.4789 - acc: 0.9565
Epoch 207/500
 - 0s - loss: 0.4687 - acc: 0.9565
Epoch 208/500
 - 0s - loss: 0.4587 - acc: 0.9565
Epoch 209/500
 - 0s - loss: 0.4489 - acc: 0.9565
Epoch 210/500
 - 0s - loss: 0.4392 - acc: 0.9565
Epoch 211/500
 - 0s - loss: 0.4298 - acc: 0.9565
Epoch 212/500
 - 0s - loss: 0.4206 - acc: 0.9565
Epoch 213/500
 - 0s - loss: 0.4116 - acc: 0.9565
Epoch 214/500
```

```
 - 0s - loss: 0.4028 - acc: 0.9565
Epoch 215/500
 - 0s - loss: 0.3942 - acc: 0.9565
Epoch 216/500
 - 0s - loss: 0.3857 - acc: 0.9565
Epoch 217/500
 - 0s - loss: 0.3775 - acc: 0.9565
Epoch 218/500
 - 0s - loss: 0.3694 - acc: 0.9565
Epoch 219/500
 - 0s - loss: 0.3616 - acc: 0.9565
Epoch 220/500
 - 0s - loss: 0.3539 - acc: 0.9565
Epoch 221/500
 - 0s - loss: 0.3464 - acc: 0.9565
Epoch 222/500
 - 0s - loss: 0.3391 - acc: 0.9565
Epoch 223/500
 - 0s - loss: 0.3320 - acc: 0.9565
Epoch 224/500
 - 0s - loss: 0.3250 - acc: 0.9565
Epoch 225/500
 - 0s - loss: 0.3183 - acc: 0.9565
Epoch 226/500
 - 0s - loss: 0.3117 - acc: 0.9565
Epoch 227/500
 - 0s - loss: 0.3052 - acc: 0.9565
Epoch 228/500
 - 0s - loss: 0.2990 - acc: 0.9565
Epoch 229/500
 - 0s - loss: 0.2929 - acc: 0.9565
Epoch 230/500
 - 0s - loss: 0.2869 - acc: 0.9565
Epoch 231/500
 - 0s - loss: 0.2811 - acc: 0.9565
Epoch 232/500
 - 0s - loss: 0.2755 - acc: 0.9565
Epoch 233/500
 - 0s - loss: 0.2701 - acc: 0.9565
Epoch 234/500
 - 0s - loss: 0.2647 - acc: 0.9565
Epoch 235/500
 - 0s - loss: 0.2596 - acc: 0.9565
Epoch 236/500
 - 0s - loss: 0.2546 - acc: 0.9565
Epoch 237/500
 - 0s - loss: 0.2497 - acc: 0.9565
Epoch 238/500
 - 0s - loss: 0.2449 - acc: 0.9565
Epoch 239/500
 - 0s - loss: 0.2404 - acc: 0.9565
Epoch 240/500
 - 0s - loss: 0.2359 - acc: 0.9565
Epoch 241/500
 - 0s - loss: 0.2316 - acc: 0.9565
Epoch 242/500
 - 0s - loss: 0.2274 - acc: 0.9565
Epoch 243/500
 - 0s - loss: 0.2233 - acc: 0.9565
Epoch 244/500
 - 0s - loss: 0.2194 - acc: 0.9565
```

```
Epoch 245/500
 - 0s - loss: 0.2155 - acc: 0.9565
Epoch 246/500
 - 0s - loss: 0.2118 - acc: 0.9565
Epoch 247/500
 - 0s - loss: 0.2082 - acc: 0.9565
Epoch 248/500
 - 0s - loss: 0.2047 - acc: 0.9565
Epoch 249/500
 - 0s - loss: 0.2013 - acc: 0.9565
Epoch 250/500
 - 0s - loss: 0.1980 - acc: 0.9565
Epoch 251/500
 - 0s - loss: 0.1948 - acc: 0.9565
Epoch 252/500
 - 0s - loss: 0.1917 - acc: 0.9565
Epoch 253/500
 - 0s - loss: 0.1886 - acc: 0.9565
Epoch 254/500
 - 0s - loss: 0.1857 - acc: 0.9565
Epoch 255/500
 - 0s - loss: 0.1828 - acc: 0.9565
Epoch 256/500
 - 0s - loss: 0.1801 - acc: 0.9565
Epoch 257/500
 - 0s - loss: 0.1774 - acc: 0.9565
Epoch 258/500
 - 0s - loss: 0.1748 - acc: 0.9565
Epoch 259/500
 - 0s - loss: 0.1723 - acc: 0.9565
Epoch 260/500
 - 0s - loss: 0.1698 - acc: 0.9565
Epoch 261/500
 - 0s - loss: 0.1674 - acc: 0.9565
Epoch 262/500
 - 0s - loss: 0.1651 - acc: 0.9565
Epoch 263/500
 - 0s - loss: 0.1629 - acc: 0.9565
Epoch 264/500
 - 0s - loss: 0.1607 - acc: 0.9565
Epoch 265/500
 - 0s - loss: 0.1586 - acc: 0.9565
Epoch 266/500
 - 0s - loss: 0.1566 - acc: 0.9565
Epoch 267/500
 - 0s - loss: 0.1546 - acc: 0.9565
Epoch 268/500
 - 0s - loss: 0.1527 - acc: 0.9565
Epoch 269/500
 - 0s - loss: 0.1509 - acc: 0.9565
Epoch 270/500
 - 0s - loss: 0.1491 - acc: 0.9565
Epoch 271/500
 - 0s - loss: 0.1473 - acc: 0.9565
Epoch 272/500
 - 0s - loss: 0.1456 - acc: 0.9565
Epoch 273/500
 - 0s - loss: 0.1440 - acc: 0.9565
Epoch 274/500
 - 0s - loss: 0.1424 - acc: 0.9565
Epoch 275/500
```

```
 - 0s - loss: 0.1408 - acc: 0.9565
Epoch 276/500
 - 0s - loss: 0.1393 - acc: 0.9565
Epoch 277/500
 - 0s - loss: 0.1378 - acc: 0.9565
Epoch 278/500
 - 0s - loss: 0.1364 - acc: 0.9565
Epoch 279/500
 - 0s - loss: 0.1350 - acc: 0.9565
Epoch 280/500
 - 0s - loss: 0.1337 - acc: 0.9565
Epoch 281/500
 - 0s - loss: 0.1323 - acc: 0.9565
Epoch 282/500
 - 0s - loss: 0.1311 - acc: 0.9565
Epoch 283/500
 - 0s - loss: 0.1298 - acc: 0.9565
Epoch 284/500
 - 0s - loss: 0.1286 - acc: 0.9565
Epoch 285/500
 - 0s - loss: 0.1275 - acc: 0.9565
Epoch 286/500
 - 0s - loss: 0.1263 - acc: 0.9565
Epoch 287/500
 - 0s - loss: 0.1252 - acc: 0.9565
Epoch 288/500
 - 0s - loss: 0.1241 - acc: 0.9565
Epoch 289/500
 - 0s - loss: 0.1231 - acc: 0.9565
Epoch 290/500
 - 0s - loss: 0.1220 - acc: 0.9565
Epoch 291/500
 - 0s - loss: 0.1210 - acc: 0.9565
Epoch 292/500
 - 0s - loss: 0.1201 - acc: 0.9565
Epoch 293/500
 - 0s - loss: 0.1191 - acc: 0.9565
Epoch 294/500
 - 0s - loss: 0.1182 - acc: 0.9565
Epoch 295/500
 - 0s - loss: 0.1173 - acc: 0.9565
Epoch 296/500
 - 0s - loss: 0.1164 - acc: 0.9565
Epoch 297/500
 - 0s - loss: 0.1156 - acc: 0.9565
Epoch 298/500
 - 0s - loss: 0.1147 - acc: 0.9565
Epoch 299/500
 - 0s - loss: 0.1139 - acc: 0.9565
Epoch 300/500
 - 0s - loss: 0.1131 - acc: 0.9565
Epoch 301/500
 - 0s - loss: 0.1123 - acc: 0.9565
Epoch 302/500
 - 0s - loss: 0.1116 - acc: 0.9565
Epoch 303/500
 - 0s - loss: 0.1108 - acc: 0.9565
Epoch 304/500
 - 0s - loss: 0.1101 - acc: 0.9565
Epoch 305/500
 - 0s - loss: 0.1094 - acc: 0.9565
```

```
Epoch 306/500
 - 0s - loss: 0.1087 - acc: 0.9565
Epoch 307/500
 - 0s - loss: 0.1081 - acc: 0.9565
Epoch 308/500
 - 0s - loss: 0.1074 - acc: 0.9565
Epoch 309/500
 - 0s - loss: 0.1067 - acc: 0.9565
Epoch 310/500
 - 0s - loss: 0.1061 - acc: 0.9565
Epoch 311/500
 - 0s - loss: 0.1055 - acc: 0.9565
Epoch 312/500
 - 0s - loss: 0.1049 - acc: 0.9565
Epoch 313/500
 - 0s - loss: 0.1043 - acc: 0.9565
Epoch 314/500
 - 0s - loss: 0.1037 - acc: 0.9565
Epoch 315/500
 - 0s - loss: 0.1032 - acc: 0.9565
Epoch 316/500
 - 0s - loss: 0.1026 - acc: 0.9565
Epoch 317/500
 - 0s - loss: 0.1021 - acc: 0.9565
Epoch 318/500
 - 0s - loss: 0.1016 - acc: 0.9565
Epoch 319/500
 - 0s - loss: 0.1010 - acc: 0.9565
Epoch 320/500
 - 0s - loss: 0.1005 - acc: 0.9565
Epoch 321/500
 - 0s - loss: 0.1000 - acc: 0.9565
Epoch 322/500
 - 0s - loss: 0.0996 - acc: 0.9565
Epoch 323/500
 - 0s - loss: 0.0991 - acc: 0.9565
Epoch 324/500
 - 0s - loss: 0.0986 - acc: 0.9565
Epoch 325/500
 - 0s - loss: 0.0982 - acc: 0.9565
Epoch 326/500
 - 0s - loss: 0.0977 - acc: 0.9565
Epoch 327/500
 - 0s - loss: 0.0973 - acc: 0.9565
Epoch 328/500
 - 0s - loss: 0.0968 - acc: 0.9565
Epoch 329/500
 - 0s - loss: 0.0964 - acc: 0.9565
Epoch 330/500
 - 0s - loss: 0.0960 - acc: 0.9565
Epoch 331/500
 - 0s - loss: 0.0956 - acc: 0.9565
Epoch 332/500
 - 0s - loss: 0.0952 - acc: 0.9565
Epoch 333/500
 - 0s - loss: 0.0948 - acc: 0.9565
Epoch 334/500
 - 0s - loss: 0.0944 - acc: 0.9565
Epoch 335/500
 - 0s - loss: 0.0940 - acc: 0.9565
Epoch 336/500
```

```
 - 0s - loss: 0.0937 - acc: 0.9565
Epoch 337/500
 - 0s - loss: 0.0933 - acc: 0.9565
Epoch 338/500
 - 0s - loss: 0.0929 - acc: 0.9565
Epoch 339/500
 - 0s - loss: 0.0926 - acc: 0.9565
Epoch 340/500
 - 0s - loss: 0.0922 - acc: 0.9565
Epoch 341/500
 - 0s - loss: 0.0919 - acc: 0.9565
Epoch 342/500
 - 0s - loss: 0.0916 - acc: 0.9565
Epoch 343/500
 - 0s - loss: 0.0912 - acc: 0.9565
Epoch 344/500
 - 0s - loss: 0.0909 - acc: 0.9565
Epoch 345/500
 - 0s - loss: 0.0906 - acc: 0.9565
Epoch 346/500
 - 0s - loss: 0.0903 - acc: 0.9565
Epoch 347/500
 - 0s - loss: 0.0900 - acc: 0.9565
Epoch 348/500
 - 0s - loss: 0.0897 - acc: 0.9565
Epoch 349/500
 - 0s - loss: 0.0894 - acc: 0.9565
Epoch 350/500
 - 0s - loss: 0.0891 - acc: 0.9565
Epoch 351/500
 - 0s - loss: 0.0888 - acc: 0.9565
Epoch 352/500
 - 0s - loss: 0.0885 - acc: 0.9565
Epoch 353/500
 - 0s - loss: 0.0882 - acc: 0.9565
Epoch 354/500
 - 0s - loss: 0.0880 - acc: 0.9565
Epoch 355/500
 - 0s - loss: 0.0877 - acc: 0.9565
Epoch 356/500
 - 0s - loss: 0.0874 - acc: 0.9565
Epoch 357/500
 - 0s - loss: 0.0872 - acc: 0.9565
Epoch 358/500
 - 0s - loss: 0.0869 - acc: 0.9565
Epoch 359/500
 - 0s - loss: 0.0867 - acc: 0.9565
Epoch 360/500
 - 0s - loss: 0.0864 - acc: 0.9565
Epoch 361/500
 - 0s - loss: 0.0862 - acc: 0.9565
Epoch 362/500
 - 0s - loss: 0.0859 - acc: 0.9565
Epoch 363/500
 - 0s - loss: 0.0857 - acc: 0.9565
Epoch 364/500
 - 0s - loss: 0.0855 - acc: 0.9565
Epoch 365/500
 - 0s - loss: 0.0852 - acc: 0.9565
Epoch 366/500
 - 0s - loss: 0.0850 - acc: 0.9565
```

```
Epoch 367/500
 - 0s - loss: 0.0848 - acc: 0.9565
Epoch 368/500
 - 0s - loss: 0.0845 - acc: 0.9565
Epoch 369/500
 - 0s - loss: 0.0843 - acc: 0.9565
Epoch 370/500
 - 0s - loss: 0.0841 - acc: 0.9565
Epoch 371/500
 - 0s - loss: 0.0839 - acc: 0.9565
Epoch 372/500
 - 0s - loss: 0.0837 - acc: 0.9565
Epoch 373/500
 - 0s - loss: 0.0835 - acc: 0.9565
Epoch 374/500
 - 0s - loss: 0.0833 - acc: 0.9565
Epoch 375/500
 - 0s - loss: 0.0831 - acc: 0.9565
Epoch 376/500
 - 0s - loss: 0.0829 - acc: 0.9565
Epoch 377/500
 - 0s - loss: 0.0827 - acc: 0.9565
Epoch 378/500
 - 0s - loss: 0.0825 - acc: 0.9565
Epoch 379/500
 - 0s - loss: 0.0823 - acc: 0.9565
Epoch 380/500
 - 0s - loss: 0.0821 - acc: 0.9565
Epoch 381/500
 - 0s - loss: 0.0820 - acc: 0.9565
Epoch 382/500
 - 0s - loss: 0.0818 - acc: 0.9565
Epoch 383/500
 - 0s - loss: 0.0816 - acc: 0.9565
Epoch 384/500
 - 0s - loss: 0.0814 - acc: 0.9565
Epoch 385/500
 - 0s - loss: 0.0813 - acc: 0.9565
Epoch 386/500
 - 0s - loss: 0.0811 - acc: 0.9565
Epoch 387/500
 - 0s - loss: 0.0809 - acc: 0.9565
Epoch 388/500
 - 0s - loss: 0.0808 - acc: 0.9565
Epoch 389/500
 - 0s - loss: 0.0806 - acc: 0.9565
Epoch 390/500
 - 0s - loss: 0.0804 - acc: 0.9565
Epoch 391/500
 - 0s - loss: 0.0803 - acc: 0.9565
Epoch 392/500
 - 0s - loss: 0.0801 - acc: 0.9565
Epoch 393/500
 - 0s - loss: 0.0800 - acc: 0.9565
Epoch 394/500
 - 0s - loss: 0.0798 - acc: 0.9565
Epoch 395/500
 - 0s - loss: 0.0797 - acc: 0.9565
Epoch 396/500
 - 0s - loss: 0.0795 - acc: 0.9565
Epoch 397/500
```

```
 - 0s - loss: 0.0794 - acc: 0.9565
Epoch 398/500
 - 0s - loss: 0.0792 - acc: 0.9565
Epoch 399/500
 - 0s - loss: 0.0791 - acc: 0.9565
Epoch 400/500
 - 0s - loss: 0.0789 - acc: 0.9565
Epoch 401/500
 - 0s - loss: 0.0788 - acc: 0.9565
Epoch 402/500
 - 0s - loss: 0.0786 - acc: 0.9565
Epoch 403/500
 - 0s - loss: 0.0785 - acc: 0.9565
Epoch 404/500
 - 0s - loss: 0.0784 - acc: 0.9565
Epoch 405/500
 - 0s - loss: 0.0782 - acc: 0.9565
Epoch 406/500
 - 0s - loss: 0.0781 - acc: 0.9565
Epoch 407/500
 - 0s - loss: 0.0780 - acc: 0.9565
Epoch 408/500
 - 0s - loss: 0.0778 - acc: 0.9565
Epoch 409/500
 - 0s - loss: 0.0777 - acc: 0.9565
Epoch 410/500
 - 0s - loss: 0.0776 - acc: 0.9565
Epoch 411/500
 - 0s - loss: 0.0775 - acc: 0.9565
Epoch 412/500
 - 0s - loss: 0.0773 - acc: 0.9565
Epoch 413/500
 - 0s - loss: 0.0772 - acc: 0.9565
Epoch 414/500
 - 0s - loss: 0.0771 - acc: 0.9565
Epoch 415/500
 - 0s - loss: 0.0770 - acc: 0.9565
Epoch 416/500
 - 0s - loss: 0.0769 - acc: 0.9565
Epoch 417/500
 - 0s - loss: 0.0768 - acc: 0.9565
Epoch 418/500
 - 0s - loss: 0.0766 - acc: 0.9565
Epoch 419/500
 - 0s - loss: 0.0765 - acc: 0.9565
Epoch 420/500
 - 0s - loss: 0.0764 - acc: 0.9565
Epoch 421/500
 - 0s - loss: 0.0763 - acc: 0.9565
Epoch 422/500
 - 0s - loss: 0.0762 - acc: 0.9565
Epoch 423/500
 - 0s - loss: 0.0761 - acc: 0.9565
Epoch 424/500
 - 0s - loss: 0.0760 - acc: 0.9565
Epoch 425/500
 - 0s - loss: 0.0759 - acc: 0.9565
Epoch 426/500
 - 0s - loss: 0.0758 - acc: 0.9565
Epoch 427/500
 - 0s - loss: 0.0757 - acc: 0.9565
```

```
Epoch 428/500
 - 0s - loss: 0.0756 - acc: 0.9565
Epoch 429/500
 - 0s - loss: 0.0754 - acc: 0.9565
Epoch 430/500
 - 0s - loss: 0.0753 - acc: 0.9565
Epoch 431/500
 - 0s - loss: 0.0752 - acc: 0.9565
Epoch 432/500
 - 0s - loss: 0.0751 - acc: 0.9565
Epoch 433/500
 - 0s - loss: 0.0751 - acc: 0.9565
Epoch 434/500
 - 0s - loss: 0.0750 - acc: 0.9565
Epoch 435/500
 - 0s - loss: 0.0749 - acc: 0.9565
Epoch 436/500
 - 0s - loss: 0.0748 - acc: 0.9565
Epoch 437/500
 - 0s - loss: 0.0747 - acc: 0.9565
Epoch 438/500
 - 0s - loss: 0.0746 - acc: 0.9565
Epoch 439/500
 - 0s - loss: 0.0745 - acc: 0.9565
Epoch 440/500
 - 0s - loss: 0.0744 - acc: 0.9565
Epoch 441/500
 - 0s - loss: 0.0743 - acc: 0.9565
Epoch 442/500
 - 0s - loss: 0.0742 - acc: 0.9565
Epoch 443/500
 - 0s - loss: 0.0741 - acc: 0.9565
Epoch 444/500
 - 0s - loss: 0.0740 - acc: 0.9565
Epoch 445/500
 - 0s - loss: 0.0740 - acc: 0.9565
Epoch 446/500
 - 0s - loss: 0.0739 - acc: 0.9565
Epoch 447/500
 - 0s - loss: 0.0738 - acc: 0.9565
Epoch 448/500
 - 0s - loss: 0.0737 - acc: 0.9565
Epoch 449/500
 - 0s - loss: 0.0736 - acc: 0.9565
Epoch 450/500
 - 0s - loss: 0.0735 - acc: 0.9565
Epoch 451/500
 - 0s - loss: 0.0735 - acc: 0.9565
Epoch 452/500
 - 0s - loss: 0.0734 - acc: 0.9565
Epoch 453/500
 - 0s - loss: 0.0733 - acc: 0.9565
Epoch 454/500
 - 0s - loss: 0.0732 - acc: 0.9565
Epoch 455/500
 - 0s - loss: 0.0731 - acc: 0.9565
Epoch 456/500
 - 0s - loss: 0.0731 - acc: 0.9565
Epoch 457/500
 - 0s - loss: 0.0730 - acc: 0.9565
Epoch 458/500
```

```
 - 0s - loss: 0.0729 - acc: 0.9565
Epoch 459/500
 - 0s - loss: 0.0728 - acc: 0.9565
Epoch 460/500
 - 0s - loss: 0.0728 - acc: 0.9565
Epoch 461/500
 - 0s - loss: 0.0727 - acc: 0.9565
Epoch 462/500
 - 0s - loss: 0.0726 - acc: 0.9565
Epoch 463/500
 - 0s - loss: 0.0726 - acc: 0.9565
Epoch 464/500
 - 0s - loss: 0.0725 - acc: 0.9565
Epoch 465/500
 - 0s - loss: 0.0724 - acc: 0.9565
Epoch 466/500
 - 0s - loss: 0.0723 - acc: 0.9565
Epoch 467/500
 - 0s - loss: 0.0723 - acc: 0.9565
Epoch 468/500
 - 0s - loss: 0.0722 - acc: 0.9565
Epoch 469/500
 - 0s - loss: 0.0721 - acc: 0.9565
Epoch 470/500
 - 0s - loss: 0.0721 - acc: 0.9565
Epoch 471/500
 - 0s - loss: 0.0720 - acc: 0.9565
Epoch 472/500
 - 0s - loss: 0.0719 - acc: 0.9565
Epoch 473/500
 - 0s - loss: 0.0719 - acc: 0.9565
Epoch 474/500
 - 0s - loss: 0.0718 - acc: 0.9565
Epoch 475/500
 - 0s - loss: 0.0717 - acc: 0.9565
Epoch 476/500
 - 0s - loss: 0.0717 - acc: 0.9565
Epoch 477/500
 - 0s - loss: 0.0716 - acc: 0.9565
Epoch 478/500
 - 0s - loss: 0.0716 - acc: 0.9565
Epoch 479/500
 - 0s - loss: 0.0715 - acc: 0.9565
Epoch 480/500
 - 0s - loss: 0.0714 - acc: 0.9565
Epoch 481/500
 - 0s - loss: 0.0714 - acc: 0.9565
Epoch 482/500
 - 0s - loss: 0.0713 - acc: 0.9565
Epoch 483/500
 - 0s - loss: 0.0713 - acc: 0.9565
Epoch 484/500
 - 0s - loss: 0.0712 - acc: 0.9565
Epoch 485/500
 - 0s - loss: 0.0711 - acc: 0.9565
Epoch 486/500
 - 0s - loss: 0.0711 - acc: 0.9565
Epoch 487/500
 - 0s - loss: 0.0710 - acc: 0.9565
Epoch 488/500
 - 0s - loss: 0.0710 - acc: 0.9565
```

```
Epoch 489/500
 - 0s - loss: 0.0709 - acc: 0.9565
Epoch 490/500
 - 0s - loss: 0.0709 - acc: 0.9565
Epoch 491/500
 - 0s - loss: 0.0708 - acc: 0.9565
Epoch 492/500
 - 0s - loss: 0.0707 - acc: 0.9565
Epoch 493/500
 - 0s - loss: 0.0707 - acc: 0.9565
Epoch 494/500
 - 0s - loss: 0.0706 - acc: 0.9565
Epoch 495/500
 - 0s - loss: 0.0706 - acc: 0.9565
Epoch 496/500
 - 0s - loss: 0.0705 - acc: 0.9565
Epoch 497/500
 - 0s - loss: 0.0705 - acc: 0.9565
Epoch 498/500
 - 0s - loss: 0.0704 - acc: 0.9565
Epoch 499/500
 - 0s - loss: 0.0704 - acc: 0.9565
Epoch 500/500
 - 0s - loss: 0.0703 - acc: 0.9565
```

Out[37]:

```
<keras.callbacks.History at 0x2e6e19b91d0>
```

In [38]:

```
# generate a sequence from a language model
def generate_seq(model, tokenizer, max_length, seed_text, n_words):
    in_text = seed_text
    # generate a fixed number of words
    for _ in range(n_words):
        # encode the text as integer
        encoded = tokenizer.texts_to_sequences([in_text])[0]
        # pre-pad sequences to a fixed length
        encoded = pad_sequences([encoded], maxlen=max_length, padding='pre')
        # predict probabilities for each word
        yhat = model.predict_classes(encoded, verbose=0)
        # map predicted word index to word
        out_word = ''
        for word, index in tokenizer.word_index.items():
            if index == yhat:
                out_word = word
                break
        # append to input
        in_text += ' ' + out_word
    return in_text
```

In [39]:

```
# evaluate model
print(generate_seq(model, tokenizer, max_length-1, 'Jack and', 5))
```

```
Jack and jill went up the hill
```

In [40]:

```
print(generate_seq(model, tokenizer, max_length-1, 'And Jill', 3))
```

```
And Jill went up the
```

In [41]:

```
print(generate_seq(model, tokenizer, max_length-1, 'fell down', 5))
```

```
fell down and broke his crown and
```

In [42]:

```
print(generate_seq(model, tokenizer, max_length-1, 'pail of', 5))
```

```
pail of water jack fell down and
```

Thank you

Shout out to our sponsor

visit their [website (http://deepanalytics.ai/)](http://deepanalytics.ai/)

Like their [Facebook page (https://www.facebook.com/DeepAnalyticsAI/)](https://www.facebook.com/DeepAnalyticsAI/)

Join our Facebook Group (https://www.facebook.com/groups/harareschoolofai/)