```
import numpy as np
from tensorflow.keras.preprocessing.text import one_hot
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Embedding
```

```
#define document
docs = ['Well done!',
        'Good work',
        'Great effort',
        'nice work',
        'Excellent!',
        'Weak',
        'Poor effort!',
        'not good',
        'poor work',
        'Could have done better.']
```

```
# define class labels
labels = np.array([1,1,1,1,1,0,0,0,0,0])
```

```
#One Hot Vector

one_hot("Excellent!",500)
```

```
[277]
```

```
#Say My vocab size is 30
#Now I want to encode all the doc
#Initially we do One-Hot Embedding
vocab_size = 30
encoded_reviews = [one_hot(d,vocab_size) for d in docs]
```

```
print((encoded_reviews))
```

```
[[22, 9], [4, 6], [15, 14], [24, 6], [21], [7], [15, 14], [29, 4], [15, 6], [18, 16, 9, 19]]
```

```
#As we can see Some Sentences are Two Word Long Some Are Three or 4
# pad documents to a max length of 4 words
max_length = 4
padded_reviews = pad_sequences(encoded_reviews,maxlen=max_length,padding='post')
print(padded_reviews)
```

```
embeded_vector_size=5
model = Sequential()
model.add(Embedding(vocab_size,4,input_length=max_length,name="embedding")) #embedding vector size
model.add(Flatten())
model.add(Dense(1,activation='sigmoid'))
```

```
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy']) #binaryCrossEntropy so output is eit
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 4, 4)              120

 flatten_1 (Flatten)         (None, 16)                0

 dense_1 (Dense)             (None, 1)                 17

=================================================================
Total params: 137
Trainable params: 137
Non-trainable params: 0
_________________________________________________________________
```

```
#train your model
model.fit(padded_reviews,labels,epochs=50,verbose=0)
```

```
<keras.callbacks.History at 0x7ff67366a6d0>
```

```
loss,accuracy = model.evaluate(padded_reviews,labels)
accuracy
```

```
1/1 [==============================] - 0s 183ms/step - loss: 0.6450 - accuracy: 0.8000
0.800000011920929
```

```
weights = model.get_layer('embedding').get_weights()[0]
len(weights)
```

```
30
```

```
weights[4]
```

```
array([ 0.08091092, -0.01694715, -0.00246512, -0.07571896], dtype=float32)
```