In [1]:
```
#Name:Priyanshu
#Roll No:60
```

In [ ]:
```
"""
Implement the Continuous Bag of Words (CBOW) Model. Stages can be:
a. Data preparation
b. Generate training data
c. Train model
d. Output
"""
```

In [4]:
```
import numpy as np
import keras.backend as K
from keras.models import Sequential
from keras.layers import Dense, Embedding, Lambda
from keras.utils import np_utils
from keras.preprocessing import sequence
from keras.preprocessing.text import Tokenizer
import gensim
data=open('C:\\Users\\hp\\Downloads\\DL Implementation\\corona.txt')
corona_data = [text for text in data if text.count(' ') >= 2]
vectorize = Tokenizer()
vectorize.fit_on_texts(corona_data)
corona_data = vectorize.texts_to_sequences(corona_data)
total_vocab = sum(len(s) for s in corona_data)
word_count = len(vectorize.word_index) + 1
window_size = 2
```

In [5]:
```
def cbow_model(data, window_size, total_vocab):
    total_length = window_size*2
    for text in data:
        text_len = len(text)
        for idx, word in enumerate(text):
            context_word = []
            target   = []
            begin = idx - window_size
            end = idx + window_size + 1
            context_word.append([text[i] for i in range(begin, end) if 0 <= i < text_len
            target.append(word)
            contextual = sequence.pad_sequences(context_word, total_length=total_length)
            final_target = np_utils.to_categorical(target, total_vocab)
            yield(contextual, final_target)
```

In [6]:
```
model = Sequential()
model.add(Embedding(input_dim=total_vocab, output_dim=100, input_length=window_size*2))
model.add(Lambda(lambda x: K.mean(x, axis=1), output_shape=(100,)))
model.add(Dense(total_vocab, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam')
for i in range(10):
    cost = 0
    for x, y in cbow_model(data, window_size, total_vocab):
        cost += model.train_on_batch(contextual, final_target)
    print(i, cost)
```

```
0 0
1 0
2 0
3 0
4 0
5 0
6 0
7 0
8 0
9 0
```

In [7]:
```
dimensions=100
vect_file = open('C:\\Users\\hp\\Downloads\\DL Implementation\\vectors.txt' ,'w')
vect_file.write('{} {}\n'.format(total_vocab,dimensions))
```

Out[7]:
```
8
```

In [8]:
```
weights = model.get_weights()[0]
for text, i in vectorize.word_index.items():
    final_vec = ' '.join(map(str, list(weights[i, :])))
    vect_file.write('{} {}\n'.format(text, final_vec))
vect_file.close()
```

In [11]:
```
cbow_output = gensim.models.KeyedVectors.load_word2vec_format('C:\\Users\\hp\\Downloads\
cbow_output.most_similar(positive=['virus'])
```

```
---------------------------------------------------------------------------
EOFError                                  Traceback (most recent call last)
Input In [11], in <cell line: 1>()
----> 1 cbow_output = gensim.models.KeyedVectors.load_word2vec_format('C:\\Users\\hp\\Downloads\\DL Implementation\\vectors.txt', binary=False)
      2 cbow_output.most_similar(positive=['virus'])

File ~\anaconda3\lib\site-packages\gensim\models\keyedvectors.py:1629, in KeyedVectors.load_word2vec_format(cls, fname, fvocab, binary, encoding, unicode_errors, limit, datatype, no_header)
   1582 @classmethod
   1583 def load_word2vec_format(
   1584     cls, fname, fvocab=None, binary=False, encoding='utf8', unicode_errors='strict',
   1585     limit=None, datatype=REAL, no_header=False,
   1586 ):
   1587     """Load KeyedVectors from a file produced by the original C word2vec-tool format.
   1588
   1589     Warnings
   (...)
   1627
   1628     """
-> 1629     return _load_word2vec_format(
   1630         cls, fname, fvocab=fvocab, binary=binary, encoding=encoding, unicode_errors=unicode_errors,
   1631         limit=limit, datatype=datatype, no_header=no_header,
   1632     )

File ~\anaconda3\lib\site-packages\gensim\models\keyedvectors.py:1976, in _load_word2vec_format(cls, fname, fvocab, binary, encoding, unicode_errors, limit, datatype, no_header, binary_chunk_size)
   1972         _word2vec_read_binary(
   1973             fin, kv, counts, vocab_size, vector_size, datatype, unicode_errors, binary_chunk_size,
   1974         )
   1975     else:
-> 1976         _word2vec_read_text(fin, kv, counts, vocab_size, vector_size, datatype, unicode_errors, encoding)
   1977 if kv.vectors.shape[0] != len(kv):
   1978     logger.info(
   1979         "duplicate words detected, shrinking matrix size from %i to %i",
   1980         kv.vectors.shape[0], len(kv),
   1981     )

File ~\anaconda3\lib\site-packages\gensim\models\keyedvectors.py:1880, in _word2vec_read_text(fin, kv, counts, vocab_size, vector_size, datatype, unicode_errors, encoding)
   1878 line = fin.readline()
   1879 if line == b'':
-> 1880     raise EOFError("unexpected end of input; is count incorrect or file otherwise damaged?")
   1881 word, weights = _word2vec_line_to_vector(line, datatype, unicode_errors, encoding)
   1882 _add_word_to_kv(kv, counts, word, weights, vocab_size)

EOFError: unexpected end of input; is count incorrect or file otherwise damaged?
```

In [ ]: