**Assignment 5**

import numpy as np

import tensorflow as tf

import keras.backend as K

from tensorflow.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('covid.txt','r')

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

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 and i != idx])

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)

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')

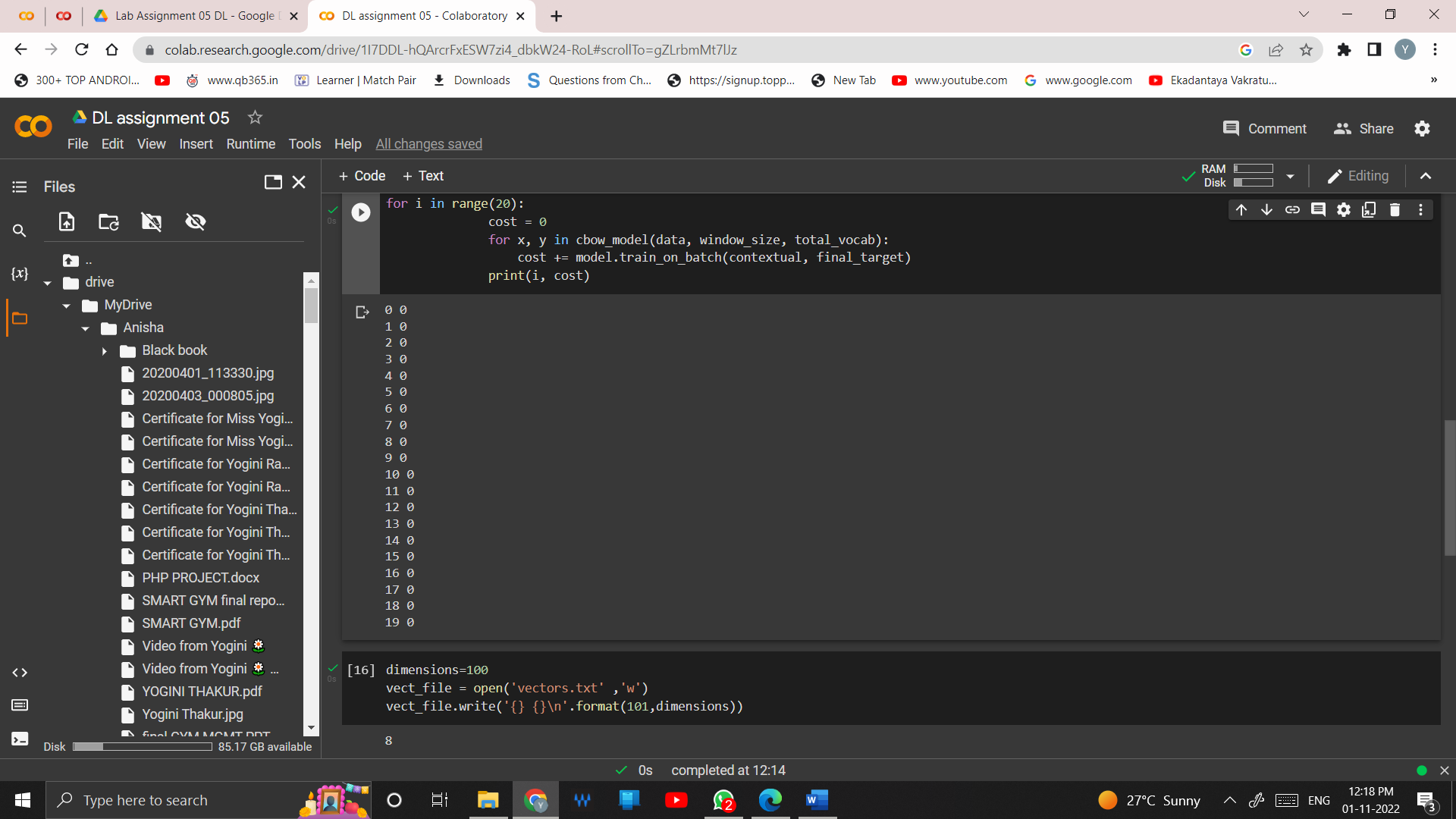
for i in range(20):

cost = 0

for x, y in cbow\_model(data, window\_size, total\_vocab):

cost += model.train\_on\_batch(contextual, final\_target)

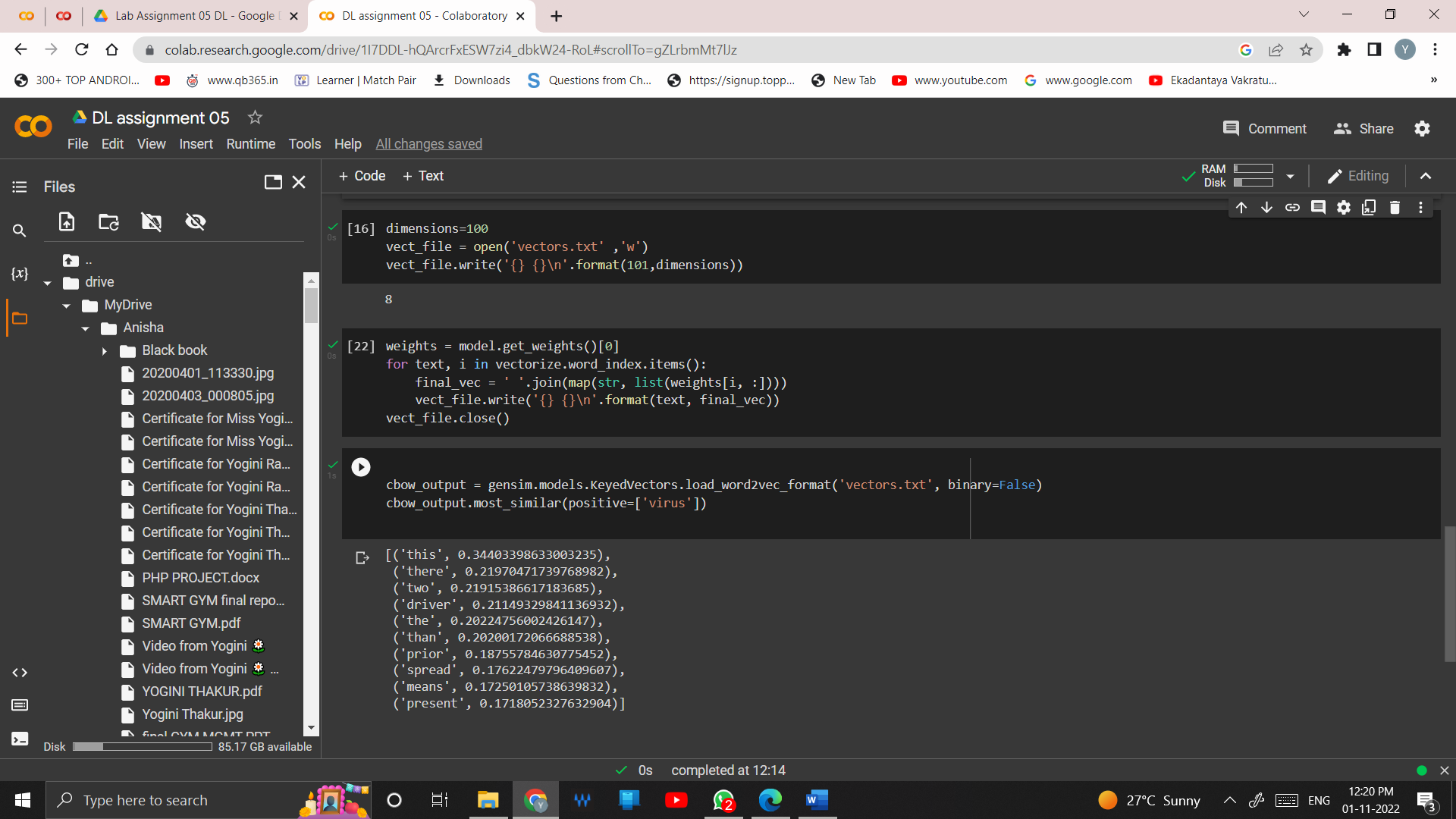
print(i, cost)



dimensions=100

vect\_file = open('vectors.txt' ,'w')

vect\_file.write('{} {}\n'.format(101,dimensions))



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()

cbow\_output = gensim.models.KeyedVectors.load\_word2vec\_format('vectors.txt', binary=False)

cbow\_output.most\_similar(positive=['virus'])

