

# Homework 4: Word Embeddings and Neural Language Models

Names: **Suzanne Becker**, Yuval Timen

# Step 1: Word2Vec paper questions

## 1. Describe how a CBOW word embedding is generated.

A CBOW word embedding can be created by using the CBOW algorithm. This would consist of training a classifier to predict a target word given some context words, and taking the weights of the trained classifier to be the embeddings. Specifically, given the sequence of length  $M$   $w_0 \dots w_M$  and a window size  $N$ , we would first create one-hot vectors for each word in the sequence. These will be used as inputs to the classifier. Then for each time step  $t$ , we will be trying to predict the word  $w_t$  given the context words within the  $N$ -sized window. We take the  $N$  words from either side of  $w_t$ , namely  $w_{t-N} \dots w_{t+N}$  excluding  $w_t$ . These will each be fed through the forward pass of the classifier, which will give an output vector of size  $1 \times V$ , representing the probability distribution of predicting the next word over our entire vocabulary. The error for each iteration in training will be the sum of the errors for all context words. We do the backward propagation, and once the weights have been updated, we move to the next timestep and shift our target word to be the next word in the sequence ( $w_{t+1}$ ), and repeat the training steps; for each timestep, we run the forward pass, find the error, and do backpropagation.

Once we complete training, our CBOW embeddings will be the weights we have learned. The weight matrix is of size  $V \times E$ , where  $E$  is the size of the embeddings we want to learn. Thus we have  $V$  vectors each of length  $E$ , meaning that for each word in our vocabulary, we've created an embedding of length  $E$ .

(We drew from source #2 to get a better understanding of how CBOW embeddings are created.)

## 2. What is a CBOW word embedding and how is it different from a skip-gram word embedding?

A CBOW word embedding is an embedding created using the CBOW algorithm - namely, embeddings learned by training a Neural Network to predict some target given some number of neighboring context words. This is very similar to the skip-gram method for generating word embeddings. The main difference is that, during the prediction task, CBOW tries to predict the target word given some neighboring context words, while skip-gram tries to predict the context words given a target word.

(this answer is also informed by source #2)

## 3. What is the task that the authors use to evaluate the generated word embeddings?

They tested that the word embeddings preserved the linear relationships between words; ideally, they wanted to have **vector("King") - vector("Man") + vector("Woman") = vector("Queen")**. They tested this by creating a large test set of 5 types of semantic relationships and 9 types of syntactic questions. This was done by manually creating a list of word pairs, for example **(Athens, Greece)** or **(great, greater)**. Then, each of these word pairs was connected to another word word pair in the same syntactic/semantic category, and the evaluation task consisted of seeing if the embeddings could accurately predict the second word in the second pair, given the relationship computed from the first pair. For example, given **(Athens, Greece) → (Oslo, Norway)**, they computed the vectors for each of the first 3 words, did the vector math, and saw whether the result was the correct last word: **vector(Athens) - vector(Greece) + vector(Oslo) =? vector(Norway)**.

(We referenced

#### 4. What are PCA and t-SNE? Why are these important to the task of training and interpreting word embeddings?

PCA (or Principal Component Analysis) is a method of finding the set of dimensions in a vector space which maximizes the variance in each dimension. This is done by taking linear combinations of our other features to try to find the first PC dimension which maximizes the variance of our data. Then, each subsequent PCA dimension is calculated using the same method, with the added condition that the new dimension must be orthogonal to all other PC dimensions. This effectively is a way of projecting the data from  $\mathbb{R}^N \rightarrow \mathbb{R}^N$ , where the new vector space's axes are sorted in order of "importance" or information. To use PCA to visualize word embeddings, we can run PCA on our embedding space, and then take the first 2 or 3 PCA dimensions and use them to visualize our embeddings in a 2 or 3 dimensional space.

On the other hand, t-SNE is a method for visualizing high-dimensional data while also trying to avoid the "curse of dimensionality". When our data lives in high-dimensional vector spaces, simply projecting the space onto a 2 or 3 dimensional plane will cause our data to overcrowd. This makes it very hard to see clusters or any other meaningful relationship between the data. t-SNE solves this problem by modeling the distribution of points in the high-dimensional space, and then re-creating that distribution in a lower dimensional space. The points in 2-d are sampled according to the distribution, meaning that the user of t-SNE gets to control how many points are sampled. This will avoid the overcrowding problem and will still allow us to visualize our high-dimensional data.

PCA and t-SNE are important in training and interpreting word embeddings because they provide a human-interpretable view on generated vector embeddings, which are otherwise long lists of numbers with no inherent semantic meaning to the human eye. Projecting vector embeddings into smaller-dimensional space would potentially reveal clusters in 2 or 3 dimensions which will allow us to visually inspect the quality of our embeddings: does each cluster contain words that we as humans would consider "similar"?

## Step 2: Train your own word embeddings

### The Spooky Author Dataset

We first examine the aptly named Spooky Authors dataset. This is a compilation of works from Edgar Allan Poe, H.P. Lovecraft, and Mary Shelley, and can be accessed on Kaggle [here \(https://www.kaggle.com/c/spooky-author-identification/overview\)](https://www.kaggle.com/c/spooky-author-identification/overview). Since we will be dealing with embeddings (ie. contexts of words), the connotation of the data is important to consider. We expect to be seeing 'spooky' bias in the embeddings to some extent.

### Our Dataset: Collected works of Nietzsche

**Describe what data set you have chosen to compare and contrast with the Spooky Authors Dataset. Make sure to describe where it comes from and its general properties.**

We chose to work with the writings of Friedrich Nietzsche, a prolific philosopher and author. We downloaded a collection of his publications from [Project Gutenberg \(https://www.gutenberg.org/ebooks/author/779\)](https://www.gutenberg.org/ebooks/author/779). Altogether, there are over 893,000 words in the corpus we collected. Nietzsche wrote a lot about human nature, life, and society, so we would expect to see greater representation of words related to being, speaking, thinking, and social organization and structure. As a philosopher, he also questions the nature of our existence and the world we have created, so his writing will also probably include more sentences structured as questions.

After downloading 14 of Nietzsche's books, we manually removed the preface metadata and any prefacing information that was not part of the original text. Then, we wrote a Python script to do the following:

- programmatically remove "footer" text (Project Gutenberg books all end with something like "\*\*\* End of this Project Gutenberg book...")
- remove chapter headings containing "Chapter", digits, or Roman numerals, using a regex
- split the text into sentences and compile all sentences from all books into a single document, `nietzsche_sentences.txt`

```
In [1]: # Imports
import csv
import re
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from gensim.models import Word2Vec
# This is a faster implementation of TSNE that Yuval found
from MulticoreTSNE import MulticoreTSNE as TSNE

%matplotlib inline
%load_ext wurlitzer

# Some constants
EMBEDDINGS_SIZE = 300
SPOOKY_DATA_PATH_TEST = "./spooky_test.csv"
SPOOKY_DATA_PATH_TRAIN = "./spooky_train.csv"
NIETZSCHE_DATA_PATH = "./nietzsche_sentences.txt"
```

```
In [2]: # Read in the data
with open(SPOOKY_DATA_PATH_TRAIN, newline='') as f:
    reader = csv.reader(f)
    reader.__next__() # skip the column names
    train = [row[1] for row in reader] # Take only the text data

with open(SPOOKY_DATA_PATH_TEST, newline='') as f:
    reader = csv.reader(f)
    reader.__next__() # skip the column names
    test = [row[1] for row in reader] # Take only the text data

# Use all the spooky data
sentences_spooky = train + test

# Read in the Nietzsche data
with open(NIETZSCHE_DATA_PATH, 'r') as f:
    # read the lines into a list
    sentences_niet = [line.strip() for line in f]
```

## Data Pre-processing

We first clean and then organize our data into a list-of-list-of strings. Each inner list represents a sentence.

The input to the data cleaning step ( `clean_data` ) is a list of sentence strings.

To clean a sentence, we first change it to entirely lowercase. Next, we remove all punctuation and numbers, replacing them with a whitespace. Then, we remove quotation marks and replace them with whitespace, being careful to preserve any single quotes that are part of contractions.

After cleaning the sentence strings, we apply Python's `.split()` operator to each sentence in the list, resulting in a list of lists of words.

```

In [3]: # Data pre-processing/cleaning

# Lowercases all words, removes punctuation,
# and replaces all instances of a number with whitespace
def clean_data(sentences):
    regex_punctuation = r'([.,;:_"\/\\\(\)\[\]!<>]|--)'
    regex_numbers = r'[0-9]'

    # Regexes matching a single quote that:
    # - precedes a word
    # - follows a word
    # - is not next to a word
    regex_single_quote_start = "([a-zA-Z])\\'([a-zA-Z])"
    regex_single_quote_end = "([a-zA-Z])\\'([a-zA-Z])"
    regex_single_quote_standalone = "([a-zA-Z])\\'([a-zA-Z])"

    output = []

    for sent in sentences:
        tmp = sent.lower()
        tmp = re.sub(regex_punctuation, ' ', tmp)
        tmp = re.sub(regex_numbers, ' ', tmp)
        tmp = re.sub(regex_single_quote_start, r'\1 \2', tmp)
        tmp = re.sub(regex_single_quote_end, r'\1 \2', tmp)
        tmp = re.sub(regex_single_quote_standalone, r'\1 \2', tmp)
        output.append(tmp)

    return output

sentences_spooky = clean_data(sentences_spooky)
sentences_niet = clean_data(sentences_niet)

# Turn each sentence into a list of words
sentences_listed_spooky = [sentence.split() for sentence in sentences_spooky]
sentences_listed_niet = [sentence.split() for sentence in sentences_niet]

```

In [ ]:

## Data Exploration & Statistics

We generate some basic statistics about the two corpora, such as sentence and word count, sentence and word length, and vocab size.

```
In [4]: # Data exploration

# Prints some basic statistics about our dataset:
# 1. Total Number of Sentences
# 2. Average Sentence Length in Words
# 3. Total Number of Words
# 4. Average Word Length in Characters
# 5. Vocabulary Size
# (The function returns the list of all words in the dataset)
def print_dataset_statistics(sentences):
    num_sentences = len(sentences)
    avg_sent_length = np.mean([len(sent.split()) for sent in sentences])
    all_words = []
    for sent in sentences:
        all_words.extend(sent.split())
    avg_word_length = np.mean([len(word) for word in all_words])
    num_words = len(all_words)
    vocab_size = len(set(all_words))

    print(f"Total sentences: {num_sentences}")
    print(f"Average sentence length: {round(avg_sent_length, 3)} words")
    print(f"Total number of words: {num_words}")
    print(f"Average word length {round(avg_word_length, 3)} characters")
    print(f"Vocabulary size: {vocab_size}")

    return all_words

print("---Spooky dataset statistics---")
all_words_spooky = print_dataset_statistics(sentences_spooky)
print("\n---Nietzsche dataset statistics---")
all_words_niet = print_dataset_statistics(sentences_niet)
```

```
---Spooky dataset statistics---
Total sentences: 27971
Average sentence length: 26.633 words
Total number of words: 744958
Average word length 4.483 characters
Vocabulary size: 29027
```

```
---Nietzsche dataset statistics---
Total sentences: 103741
Average sentence length: 8.615 words
Total number of words: 893702
Average word length 4.647 characters
Vocabulary size: 31371
```

## a) Train embedding on GIVEN dataset

```
In [5]: # Create the embeddings, save the KeyedVector, toss the model
model_spooky = Word2Vec(sentences=sentences_listed_spooky, size=EMBEDDING_SIZE,
                        sg=1, window=5, min_count=1)
spooky_embeddings = model_spooky.wv
del model_spooky
```

## b) Train embedding on YOUR dataset

```
In [6]: model_niet = Word2Vec(sentences=sentences_listed_niet, size=EMBEDDINGS_SIZE, sg=1, window=5, min_count=1)
niet_embeddings = model_niet.wv
del model_niet
```

## Step 3: Evaluate the differences between the word embeddings

(make sure to include graphs, figures, and paragraphs with full sentences)



```

In [7]: # This function prints some quick similarities and comparisons for the g
        iven embeddings
        def quick_evaluate_embeddings(embeddings):

            similar_to_king = embeddings.most_similar('king')[0][0]
            similar_to_horse = embeddings.most_similar('horse')[0][0]
            similar_to_man = embeddings.most_similar('man')[0][0]
            king_minus_man_plus_woman = embeddings.most_similar(positive=['king'
, 'woman'], negative=['man'])[0][0]
            odd_one_out_food = embeddings.doesnt_match(('bread', 'meat', 'vegeta
bles', 'peasant'))
            odd_one_out_days = embeddings.doesnt_match(('monday', 'wednesday',
'friday', 'favorite'))
            odd_one_out_animals = embeddings.doesnt_match(('cat', 'dog', 'sheep'
, 'wolf', 'hose'))

            print(f"Word most similar to 'king': {similar_to_king}")
            print(f"Word most similar to 'horse': {similar_to_horse}")
            print(f"Word most similar to 'man': {similar_to_man}")
            print(f"King - Man + Woman: {king_minus_man_plus_woman}")
            print(f"Was the model able to distinguish 'peasant' from food? {odd_
one_out_food == 'peasant'}")
            print(f"Was the model able to distinguish 'favorite' from weekdays?
{odd_one_out_days == 'favorite'}")
            print(f"Was the model able to distinguish 'train' from animals? {odd
_one_out_animals == 'train'}")

        print('Output for Spooky Embeddings:')
        quick_evaluate_embeddings(spooky_embeddings)
        print('-----')
        print('Output for Nietzsche Embeddings')
        quick_evaluate_embeddings(niet_embeddings)

```

Output for Spooky Embeddings:  
Word most similar to 'king': visitor  
Word most similar to 'horse': coat  
Word most similar to 'man': woman  
King - Man + Woman: mother's  
Was the model able to distinguish 'peasant' from food? True  
Was the model able to distinguish 'favorite' from weekdays? True  
Was the model able to distinguish 'train' from animals? False  
-----

Output for Nietzsche Embeddings  
Word most similar to 'king': buddha  
Word most similar to 'horse': carrieth  
Word most similar to 'man': priest  
King - Man + Woman: ugliest  
Was the model able to distinguish 'peasant' from food? False  
Was the model able to distinguish 'favorite' from weekdays? True  
Was the model able to distinguish 'train' from animals? False

/Users/suzanne/opt/anaconda3/lib/python3.8/site-packages/gensim/models/  
keyedvectors.py:877: FutureWarning: arrays to stack must be passed as a  
"sequence" type such as list or tuple. Support for non-sequence iterabl  
es such as generators is deprecated as of NumPy 1.16 and will raise an  
error in the future.

```
    vectors = vstack(self.word_vec(word, use_norm=True) for word in used_  
words).astype(REAL)
```

```

In [8]: # Building the t-SNE projection

# This is the V x E matrix; the full-length embedding for each vocab word
# Each unique word $w_{i}$ represents the row $V_{i}$
# Each row is a vector of length E which corresponds to that word's embedding
embedding_matrix_spooky = spooky_embeddings[spooky_embeddings.vocab]
embedding_matrix_niet = niet_embeddings[niet_embeddings.vocab]
print(f"Original shape of the spooky matrix: {embedding_matrix_spooky.shape}") # (31851, 300)
print(f"Original shape of the philosophical matrix: {embedding_matrix_niet.shape}") # (31851, 300)


# We first run PCA and keep the first num_components dimensions before we pass into t-SNE
num_components = 50
random_seed = 100

pca_spooky = PCA(n_components=num_components)
pca_embeddings_spooky = pca_spooky.fit_transform(embedding_matrix_spooky)
pca_niet = PCA(n_components=num_components)
pca_embeddings_niet = pca_niet.fit_transform(embedding_matrix_niet)
print(f"New shape of the spooky matrix: {pca_embeddings_spooky.shape}") # (31851, 50)
print(f"New shape of the philosophical matrix: {pca_embeddings_niet.shape}") # (31851, 50)


# Since we ran PCA and kept only the first few dimensions, we want to know how much
# of the original information is still preserved in this projection. The concept of
# "explained variance" is helpful here. Our original 300-dimensional vector space has
# a certain amount of variance among the data. This is our "total variance". When we
# project to 50 dimensions, we lose some of that variance, since we lose dimensions along
# which our data can vary. We calculate the total amount of variance in our 50-dimensional
# space and divide that number by our total variance. This yields the amount of variance
# preserved after our projection. So while our data might not have *all* of the original
# variance, we can still see that it has a significant percentage of the total variance, and
# so we only lose a small amount of information while reducing our space a lot. Thanks PCA!
print(f"SPOOKY: Cumulative explained variance of first {num_components} PC dimensions: {100 * round(np.sum(pca_spooky.explained_variance_ratio_), 3)}%")
print(f"PHILOSOPHICAL: Cumulative explained variance of first {num_components} PC dimensions: {100 * round(np.sum(pca_niet.explained_variance_ratio_), 3)}%")

```

Original shape of the spooky matrix: (29027, 300)  
Original shape of the philosophical matrix: (31371, 300)  
New shape of the spooky matrix: (29027, 50)  
New shape of the philosophical matrix: (31371, 50)  
SPOOKY: Cumulative explained variance of first 50 PC dimensions: 99.199  
99837875366%  
PHILOSOPHICAL: Cumulative explained variance of first 50 PC dimensions:  
98.19999933242798%

```
In [9]: # Now we run t-SNE
tsne_spooky = TSNE(n_components=2, perplexity=45, verbose=True, random_s
tate=random_seed, n_iter=500)
tsne_embeddings_spooky = tsne_spooky.fit_transform(pca_embeddings_spooky
)
```

Performing t-SNE using 1 cores.  
Using no\_dims = 2, perplexity = 45.000000, and theta = 0.500000  
Computing input similarities...  
Building tree...  
- point 2902 of 29027  
- point 5804 of 29027  
- point 8706 of 29027  
- point 11608 of 29027  
- point 14510 of 29027  
- point 17412 of 29027  
- point 20314 of 29027  
- point 23216 of 29027  
- point 26118 of 29027  
- point 29020 of 29027  
Done in 11.00 seconds (sparsity = 0.007387)!  
Learning embedding...  
Iteration 51: error is 103.172970 (50 iterations in 24.00 seconds)  
Iteration 101: error is 95.869392 (50 iterations in 18.00 seconds)  
Iteration 151: error is 89.486511 (50 iterations in 16.00 seconds)  
Iteration 201: error is 88.339251 (50 iterations in 15.00 seconds)  
Iteration 251: error is 88.051582 (50 iterations in 15.00 seconds)  
Iteration 301: error is 3.745427 (50 iterations in 15.00 seconds)  
Iteration 351: error is 3.470659 (50 iterations in 13.00 seconds)  
Iteration 401: error is 3.320250 (50 iterations in 18.00 seconds)  
Iteration 451: error is 3.219545 (50 iterations in 17.00 seconds)  
Iteration 500: error is 3.146376 (50 iterations in 14.00 seconds)  
Fitting performed in 165.00 seconds.

```
In [10]: # Now we run t-SNE (again)
         tsne_niet = TSNE(n_components=2, perplexity=45, verbose=True, random_state=random_seed, n_iter=500)
         tsne_embeddings_niet = tsne_niet.fit_transform(pca_embeddings_niet)
```

Performing t-SNE using 1 cores.

Using no\_dims = 2, perplexity = 45.000000, and theta = 0.500000

Computing input similarities...

Building tree...

- point 3137 of 31371
- point 6274 of 31371
- point 9411 of 31371
- point 12548 of 31371
- point 15685 of 31371
- point 18822 of 31371
- point 21959 of 31371
- point 25096 of 31371
- point 28233 of 31371
- point 31370 of 31371

Done in 14.00 seconds (sparsity = 0.007016)!

Learning embedding...

Iteration 51: error is 103.449589 (50 iterations in 23.00 seconds)

Iteration 101: error is 94.671524 (50 iterations in 20.00 seconds)

Iteration 151: error is 89.484101 (50 iterations in 17.00 seconds)

Iteration 201: error is 88.073050 (50 iterations in 20.00 seconds)

Iteration 251: error is 87.629793 (50 iterations in 23.00 seconds)

Iteration 301: error is 3.884314 (50 iterations in 20.00 seconds)

Iteration 351: error is 3.625237 (50 iterations in 18.00 seconds)

Iteration 401: error is 3.472446 (50 iterations in 18.00 seconds)

Iteration 451: error is 3.368306 (50 iterations in 18.00 seconds)

Iteration 500: error is 3.294004 (50 iterations in 18.00 seconds)

Fitting performed in 195.00 seconds.

```

In [11]: # Visualizing the data

vis_x_spooky = tsne_embeddings_spooky[:, 0]
vis_y_spooky = tsne_embeddings_spooky[:, 1]

vis_x_niet = tsne_embeddings_niet[:, 0]
vis_y_niet = tsne_embeddings_niet[:, 1]

fig, (ax1, ax2) = plt.subplots(1, 2, sharey=True, figsize=(16, 10))

ax1.scatter(vis_x_spooky, vis_y_spooky, c='grey', marker='.')
ax2.scatter(vis_x_niet, vis_y_niet, c='grey', marker='.')

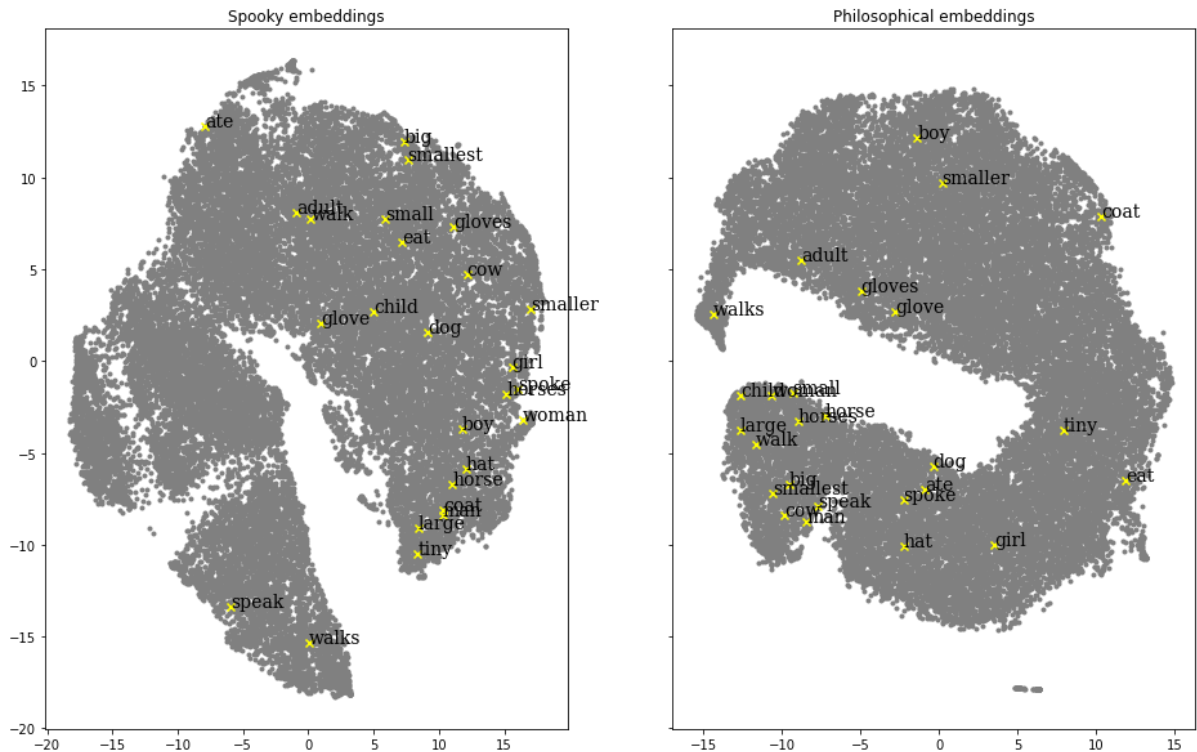
def visualize_clusters(current_ax, embeddings, tsne_model):
    # We highlight the location of several words of interest on this chart.
    words_of_interest = ('man', 'woman', 'boy', 'girl', 'child', 'adult',
        ,
                        'coat', 'hat', 'glove', 'gloves',
                        'dog', 'horse', 'horses', 'cow',
                        'large', 'big', 'tiny', 'small', 'smaller', 'smallest',
                        'walk', 'walks', 'speak', 'spoke', 'eat', 'ate')
    vectors_of_interest = []
    # Need to look up the index of the word to find its corresponding t-SNE vector
    for word in words_of_interest:
        idx_of_word = embeddings.vocab[word].index
        vectors_of_interest.append(tsne_model.embedding[idx_of_word])

    # Text styling parameters
    font= {'family':'serif','color':'black','weight':'normal','size':14}

    # Draw the words of interest on and label them
    for idx, vec in enumerate(vectors_of_interest):
        current_ax.scatter(vec[0], vec[1], c='yellow', marker='x')
        current_ax.text(vec[0], vec[1], words_of_interest[idx], fontdict=font)

visualize_clusters(ax1, spooky_embeddings, tsne_spooky)
visualize_clusters(ax2, niet_embeddings, tsne_niet)
ax1.set_title('Spooky embeddings')
ax2.set_title('Philosophical embeddings')
plt.show()

```



## Step 4: Feedforward Neural Language Model

### a) First, encode your text into integers

```
In [12]: # Importing utility functions from Keras
from keras.preprocessing.text import Tokenizer
from keras.utils import to_categorical
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Embedding

NGRAM = 4 # The size of the ngram language model we will train
```

```
In [13]: # First, we map each word to a unique index
tokenizer_spooky = Tokenizer()
tokenizer_spooky.fit_on_texts(sentences_listed_spooky)
sentences_listed_unique_id_spooky = tokenizer_spooky.texts_to_sequences(
    sentences_listed_spooky)

tokenizer_niet = Tokenizer()
tokenizer_niet.fit_on_texts(sentences_listed_niet)
sentences_listed_unique_id_niet = tokenizer_niet.texts_to_sequences(
    sentences_listed_niet)
```

In [ ]:

```

In [14]: # This function will return a single numpy array of
# length (N-1)xE, where N is the n-gram order (implicitly, the length of
# our iterable),
# and E is the length of our embedding. The output is a concatenation of
# the
# word embeddings in our iterable.
def concat_word_embeddings(iterable, embeddings):
    disjoint_embeddings = [spooky_embeddings.get_vector(word) for word in
iterable]
    return np.concatenate(disjoint_embeddings)

mini_concat = ['did', 'you', 'win']

# We expect the resulting shape to be len(mini_concat) * EMBEDDINGS_SIZE
print("Resulting shape from concatenating the word embeddings for mini-c
oncat:")
print(concat_word_embeddings(mini_concat, spooky_embeddings).shape)
print("Expected shape from concatenating the word embeddings for mini-co
ncat:")
print((len(mini_concat) * EMBEDDINGS_SIZE,))

```

```

Resulting shape from concatenating the word embeddings for mini-concat:
(900,)
Expected shape from concatenating the word embeddings for mini-concat:
(900,)

```



```

In [15]: # We format our data to be passed into the model

# This function will take our encoded list of lists
# It will return a generator which yields values like ([[x1, x2, ... ,
#           x(n-1)], y) ...]
# In this case, each X_i will be the word embedding corresponding to tha
# t word
def data_to_xy_tuples(data, n_gram, tokenizer, embeddings):
    num_classes = max(tokenizer.word_index.values()) + 1
    for sentence in data:
        if len(sentence) < n_gram:
            continue
        for i in range(len(sentence) - n_gram):
            # yield sentence[i : n_gram + i], sentence[n_gram + i]
            yield (concat_word_embeddings(sentence[i : n_gram + i], embeddings),
                  to_categorical(tokenizer.texts_to_sequences([sentence
[n_gram + i]]), num_classes=num_classes))

# Test that data_to_xy_tuples works properly
mini_test = [['welcome', 'back'], ['hope', 'you', 'had', 'a', 'good', 'weekend']]
tok = Tokenizer()
tok.fit_on_texts(mini_test)

indexed = tok.texts_to_sequences(mini_test)
print(indexed)
one_hot = [to_categorical(i) for i in indexed]
print(one_hot)

print("Result of the test sequence being separated into tuples")
for x, y in data_to_xy_tuples(mini_test, 3, tok, spooky_embeddings):
    print(f"{x} -> {y}")

# Finally, let's prepare our data sequence to be fed into the model
data_generator_spooky = data_to_xy_tuples(sentences_listed_spooky, NGRAM,
, tokenizer_spooky, spooky_embeddings)
data_generator_niet = data_to_xy_tuples(sentences_listed_niet, NGRAM, tokenizer_niet,
niet_embeddings)

```

```

[[1, 2], [3, 4, 5, 6, 7, 8]]
[array([[0., 1., 0.],
        [0., 0., 1.]], dtype=float32), array([[0., 0., 0., 1., 0., 0.,
0., 0., 0.],
        [0., 0., 0., 0., 1., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 1., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 1., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 1.]], dtype=float32)]

```

Result of the test sequence being separated into tuples

```

[-6.77222535e-02  1.04836203e-01 -1.67276874e-01 -7.91453123e-02
-9.37306136e-02 -1.17891161e-02 -2.14642704e-01  4.95134801e-01
-7.51628652e-02 -9.15366318e-03 -1.89321086e-01 -1.10866904e-01
-3.21907341e-01  1.28275767e-01  1.91212058e-01  1.39977321e-01
-3.57946664e-01 -8.99181291e-02 -9.39083565e-03  1.14940912e-01
-1.23061225e-01  1.53338164e-01 -8.05286244e-02 -6.29154369e-02
-1.25321880e-01 -2.45987754e-02 -2.54349202e-01  3.31839956e-02
 1.37205434e-03 -5.27702505e-03  1.19366750e-01  5.41815199e-02
 3.23370159e-01  7.39920065e-02 -2.87569940e-01 -3.35070550e-01
-1.33938864e-01  1.45502776e-01 -2.15152949e-02 -4.02066894e-02
-5.41693941e-02 -4.76312488e-01  6.41311472e-03  1.16651446e-01
 1.73658319e-02 -6.18428476e-02 -1.75055206e-01  1.97208032e-01
 2.70620823e-01 -1.40106976e-01 -2.13293463e-01  5.49783595e-02
 1.00377975e-02  4.64082509e-02  2.80285984e-01 -2.45584011e-01
 4.02436927e-02  9.31063667e-02  6.08814247e-02 -3.72353685e-03
-3.29528064e-01  3.67469825e-02  2.46753961e-01 -4.08870354e-03
-3.88571471e-02  4.47473899e-02  7.16208965e-02 -2.41293401e-01
 2.41930887e-01 -1.17930062e-02  1.17658801e-01 -1.87156379e-01
 1.43700186e-02 -1.74136143e-02  1.18723921e-01  1.31567925e-01
 1.15636535e-01  1.62533551e-01  1.90862775e-01 -9.78561938e-02
-1.40970901e-01 -9.37454682e-03  1.23627990e-01  6.88956007e-02
 1.31524369e-01  3.07763457e-01 -3.62294406e-01  7.61491852e-03
-9.12164822e-02  1.14875242e-01 -6.81105927e-02 -2.37334985e-02
 2.52381563e-02 -2.48981845e-02  1.71859518e-01  1.51391372e-01
-1.69868302e-02  3.21864307e-01 -1.18303411e-01  1.27043754e-01
-2.74147671e-02  1.52728111e-01  2.17707276e-01  3.68616849e-01
-5.23185432e-02  7.51198530e-02  3.66854146e-02 -3.12457085e-01
-1.83309570e-01 -1.74441472e-01  4.48806919e-02  1.71799704e-01
-3.87410969e-01 -1.30381351e-02  2.30750024e-01  1.76460132e-01
 7.51043558e-02  4.32585692e-03 -1.21512204e-01 -1.25759859e-02
 2.76087727e-02  8.26792791e-02  5.61439097e-02 -5.39707132e-02
 2.99327429e-02  7.31965452e-02 -3.09516579e-01  1.02532946e-01
-1.35273203e-01 -4.59859595e-02  2.11088546e-02  2.49780610e-01
 3.59052181e-01  2.76720580e-02  5.57942130e-02 -5.37455343e-02
-4.95811440e-02 -5.14847599e-02  4.23172163e-03 -5.33863232e-02
-2.55845606e-01 -2.29451090e-01  1.29503503e-01 -1.74233302e-01
-8.15605465e-03  3.74196738e-01  1.45282075e-01 -3.60030591e-01
-8.79246369e-02  7.70264864e-02 -3.25797051e-01  2.82346636e-01
-1.35168791e-01 -1.80054322e-01 -1.84526294e-01 -8.83302763e-02
-1.16242141e-01  1.39005512e-01  1.51327223e-01 -1.55225277e-01
 2.73754634e-02 -1.34796366e-01 -7.03550801e-02  2.83436216e-02
-3.83124314e-02  1.17157377e-01 -1.37998313e-01  1.04963137e-02
-2.34285221e-01 -9.43258181e-02 -8.61612782e-02 -3.60785127e-02
-3.43452021e-02  6.24632947e-02  2.38367751e-01 -1.55950025e-01
-3.72287869e-01  2.42696747e-01  2.40253255e-01  2.47557193e-01
 2.51589110e-03 -5.21625392e-02 -1.68330014e-01  3.30839485e-01
 5.26654348e-02 -4.10261787e-02  3.73909473e-02  8.98618698e-02

```

|                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|
| -4.16740216e-02 | 1.03158712e-01  | 1.50262779e-02  | 2.22399652e-01  |
| 2.59734869e-01  | -3.43729705e-01 | -9.66755599e-02 | 1.61839038e-01  |
| -2.17624560e-01 | -3.35533470e-01 | -1.43709049e-01 | 1.34352539e-02  |
| -3.13880950e-01 | 4.22078837e-03  | 1.26687109e-01  | 1.51474014e-01  |
| 8.43822360e-02  | 2.55075991e-01  | -3.14639837e-01 | 1.23054028e-01  |
| 2.71835085e-02  | -2.53995508e-01 | -4.99242246e-01 | -1.49485990e-01 |
| -3.38908553e-01 | -2.31177241e-01 | -1.99075714e-02 | -3.02206159e-01 |
| -4.53221709e-01 | 4.99357432e-02  | -2.22025663e-02 | -8.34789798e-02 |
| 3.99760194e-02  | 1.05902873e-01  | 1.13190554e-01  | -1.81578934e-01 |
| -2.82714248e-01 | -1.33686706e-01 | -1.14826761e-01 | 1.92009762e-01  |
| 1.25316769e-01  | -1.80840999e-01 | 3.02086212e-02  | -2.18837604e-01 |
| 2.14292794e-01  | -3.48659426e-01 | -2.14693889e-01 | -1.49142575e-02 |
| 6.75693825e-02  | 1.52531147e-01  | -1.65614218e-01 | -1.63466915e-01 |
| 1.68187432e-02  | 9.80993062e-02  | -3.65543485e-01 | 7.00649098e-02  |
| -5.78065142e-02 | 6.69743717e-02  | -1.99401438e-01 | 7.63950050e-02  |
| -1.33814037e-01 | 4.69109528e-02  | -6.93249777e-02 | -4.44345400e-02 |
| -1.08115450e-01 | 2.38453358e-01  | -1.87085010e-03 | -2.44124591e-01 |
| -2.54558563e-01 | 5.27853258e-02  | -2.26343825e-01 | 2.77329236e-01  |
| 1.98724851e-01  | -1.08570717e-01 | -1.60318092e-01 | 8.42818469e-02  |
| -3.42211984e-02 | -9.79630128e-02 | -8.91339034e-02 | -8.19835439e-02 |
| -4.91180383e-02 | 7.23313764e-02  | -2.84499347e-01 | -7.96373114e-02 |
| -1.25435233e-01 | 7.15124086e-02  | 4.04090226e-01  | 1.54164359e-01  |
| -1.60913378e-01 | -2.79379040e-01 | -2.51308195e-02 | 9.67500955e-02  |
| -2.76564658e-01 | 2.35457256e-01  | 3.79704498e-02  | 1.69491656e-02  |
| -1.05753943e-01 | 2.70522833e-02  | -1.03395090e-01 | 3.44682969e-02  |
| -3.35137367e-01 | -1.50062004e-02 | 6.55504838e-02  | -3.61299999e-02 |
| -2.45395586e-01 | 3.30482014e-02  | -7.31515586e-02 | 1.10669538e-01  |
| 1.09132238e-01  | -3.38715538e-02 | -2.12810829e-01 | 2.28354353e-02  |
| -8.19788314e-03 | 1.93318456e-01  | -2.55737096e-01 | -5.85262999e-02 |
| -1.12143733e-01 | 1.28226191e-01  | -1.63523600e-01 | 3.29153985e-01  |
| 1.72688179e-02  | -7.23662078e-02 | -4.75730926e-01 | -1.56640321e-01 |
| -3.25731456e-01 | 6.37583286e-02  | -3.48807164e-02 | 7.54046515e-02  |
| -2.67262399e-01 | -2.07818210e-01 | -4.25361954e-02 | 1.84763186e-02  |
| -2.99307972e-01 | 1.15987822e-01  | -1.50392756e-01 | -8.54285732e-02 |
| -1.55638503e-02 | -1.67371407e-01 | 5.10390336e-03  | 9.89027023e-02  |
| -2.01399982e-01 | 3.10537755e-01  | 1.34452224e-01  | 1.30464569e-01  |
| 1.95780411e-01  | 5.01945540e-02  | -5.31948149e-01 | -1.85767025e-01 |
| 1.22942381e-01  | 7.54639953e-02  | -1.22166574e-01 | -4.99696508e-02 |
| -6.83552548e-02 | -6.58104420e-01 | 6.36971444e-02  | 1.41285956e-01  |
| -1.44623920e-01 | -8.98232311e-03 | -3.38681191e-01 | 3.19188088e-01  |
| 3.95171702e-01  | -5.40090322e-01 | -5.58430795e-04 | 7.72019476e-02  |
| 3.45504552e-01  | 2.31432348e-01  | 5.54515064e-01  | -3.62869292e-01 |
| -1.29860565e-01 | 5.47985673e-01  | 6.30740598e-02  | 8.19746926e-02  |
| -1.09874740e-01 | 2.18467295e-01  | -3.48884016e-02 | 2.04096988e-01  |
| -4.61730473e-02 | 1.16994128e-01  | -1.67139530e-01 | -9.06317532e-02 |
| 1.63974106e-01  | -2.07277089e-01 | -9.52801108e-02 | -1.60821632e-01 |
| 1.71140358e-01  | 1.60285488e-01  | 2.45889872e-01  | 6.92768022e-02  |
| -7.31295496e-02 | 2.27498814e-01  | 1.54240772e-01  | -3.36584568e-01 |
| 2.05547854e-01  | 9.88019481e-02  | 7.80080399e-03  | 2.23548993e-01  |
| 3.24082553e-01  | 2.67415315e-01  | -4.23092872e-01 | -1.86155140e-01 |
| -5.90210967e-02 | 1.89443082e-01  | -2.48147652e-01 | 1.30945921e-03  |
| 7.92904645e-02  | -9.25080553e-02 | 2.97049358e-02  | 1.24130942e-01  |
| -4.43422854e-01 | 3.02665323e-01  | -2.96746910e-01 | 2.79361755e-01  |
| -1.00608163e-01 | 3.18943679e-01  | 2.59121776e-01  | 2.07691684e-01  |
| -3.09031337e-01 | 2.35513002e-01  | -4.21084911e-02 | -2.48895317e-01 |
| -9.59007740e-02 | -3.03914875e-01 | 1.31026596e-01  | 3.34041193e-02  |
| -4.23344195e-01 | 2.01583043e-01  | 2.56819606e-01  | 1.58521101e-01  |

|                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|
| 9.14690793e-02  | 2.74726987e-01  | -4.80642915e-02 | -2.05094770e-01 |
| 4.60502543e-02  | 5.64997410e-03  | 6.29949346e-02  | -1.33641645e-01 |
| -1.22518921e-02 | -2.19497427e-01 | -4.34247665e-02 | 2.12650120e-01  |
| -2.38290668e-01 | 1.47572299e-02  | 2.08088800e-01  | 1.95156559e-01  |
| 2.76555806e-01  | -2.51404196e-02 | -1.40751600e-01 | 1.54192463e-01  |
| -1.01445548e-01 | 6.77790865e-02  | 1.42202742e-04  | -3.94800723e-01 |
| -2.78166771e-01 | -2.08751321e-01 | 4.67118472e-01  | 6.68946281e-02  |
| -1.24982662e-01 | 5.39936900e-01  | 2.07879215e-01  | -6.27942264e-01 |
| -5.96871190e-02 | 2.15298489e-01  | -2.51414299e-01 | 8.56768340e-02  |
| -1.22809909e-01 | 3.64094041e-02  | -1.87768579e-01 | 6.21910393e-02  |
| -8.15605223e-02 | 1.15067698e-01  | 2.27230117e-02  | -7.60399923e-02 |
| 9.59183797e-02  | -2.26689413e-01 | -3.21219891e-01 | 2.80338734e-01  |
| -3.12000751e-01 | 7.85248056e-02  | -4.90650646e-02 | -1.91503912e-01 |
| -2.25585863e-01 | -3.92571986e-01 | -1.40992152e-02 | -1.17863886e-01 |
| 8.60270932e-02  | 7.31864152e-03  | 4.06437427e-01  | -1.05658554e-01 |
| -3.58285278e-01 | 3.86391789e-01  | -8.41783639e-03 | -1.05645679e-01 |
| -6.26050457e-02 | -3.01659763e-01 | 1.03028394e-01  | 5.19222438e-01  |
| 1.95978776e-01  | -1.01899147e-01 | 3.13420415e-01  | 5.11938557e-02  |
| 2.95158833e-01  | -3.36767063e-02 | 4.48371470e-02  | 5.21192193e-01  |
| 2.78637022e-01  | -4.09705877e-01 | 3.65453474e-02  | 2.09444687e-01  |
| -2.35680997e-01 | -5.46684444e-01 | -6.87311217e-03 | 2.94725299e-01  |
| -5.34977615e-01 | -1.52939141e-01 | 1.93228826e-01  | 2.54261613e-01  |
| 1.77904129e-01  | 1.94128066e-01  | -4.10632163e-01 | 3.25125724e-01  |
| 2.31717136e-02  | -4.24731374e-01 | -5.85962355e-01 | 2.19214670e-02  |
| -2.25591764e-01 | -2.19828025e-01 | -2.28205118e-02 | -3.72632384e-01 |
| -2.67896384e-01 | -1.27756312e-01 | 1.93230763e-01  | -2.01365992e-01 |
| 5.95798641e-02  | 2.77440786e-01  | 8.24818164e-02  | -1.47691175e-01 |
| -5.18139005e-01 | -2.19808012e-01 | 1.38957992e-01  | 1.38900071e-01  |
| -9.44016427e-02 | -2.24219382e-01 | 2.07139924e-01  | -2.33354717e-01 |
| -5.80870919e-02 | -1.57668680e-01 | -7.70893544e-02 | 7.81492069e-02  |
| -6.59244657e-02 | 1.69186592e-01  | -1.44687667e-01 | -2.11417049e-01 |
| -7.01892823e-02 | -3.96145061e-02 | -5.46510458e-01 | 1.28850058e-01  |
| 2.22768903e-01  | -2.98677236e-01 | -1.18267253e-01 | 6.33210540e-02  |
| -5.74567914e-02 | -1.67560130e-01 | -1.39726549e-01 | 2.93317251e-02  |
| 1.06791906e-01  | -5.84209077e-02 | 6.79801702e-02  | -2.59042174e-01 |
| -2.82391757e-01 | -2.19578519e-01 | -3.81942928e-01 | 2.21975192e-01  |
| 3.05138379e-02  | 1.92736208e-01  | -1.85888261e-01 | -1.06375091e-01 |
| -2.32108030e-02 | -6.32791370e-02 | -1.58603340e-01 | 9.82918311e-03  |
| -1.30309612e-01 | -1.20525494e-01 | -4.22838628e-01 | -2.88072854e-01 |
| -5.22131622e-02 | -4.51118201e-02 | 2.37653092e-01  | 6.80113658e-02  |
| -5.76707199e-02 | -4.53665197e-01 | 1.98694944e-01  | 6.51667640e-02  |
| -3.39328945e-01 | 3.13593984e-01  | 8.70105624e-02  | -1.60077110e-01 |
| -1.27475768e-01 | -4.41596538e-01 | 7.60736838e-02  | -1.28953993e-01 |
| -3.30101579e-01 | -2.48410963e-02 | -2.69903317e-02 | -3.11167035e-02 |
| -1.59395128e-01 | 1.90406851e-02  | -1.47860155e-01 | 7.09712058e-02  |
| 5.60208082e-01  | 1.01584129e-01  | -3.00363451e-01 | -1.45654321e-01 |
| -1.28002688e-01 | -1.52486414e-01 | -3.12842727e-01 | 4.64192731e-03  |
| -2.96131939e-01 | -1.98212892e-01 | -1.51366189e-01 | 2.97824711e-01  |
| -2.23264307e-01 | 8.09413791e-02  | -2.10164845e-01 | -1.84343770e-01 |
| -1.78107738e-01 | -1.64515167e-01 | 4.20021296e-01  | 5.25286682e-02  |
| 2.76020393e-02  | 1.17821638e-02  | -9.70714539e-02 | -6.89446330e-02 |
| 2.81050026e-01  | 1.86055824e-02  | 1.15677491e-01  | 9.43794497e-04  |
| -1.11368045e-01 | -1.97569832e-01 | -1.38451830e-01 | 5.19487299e-02  |
| -2.94302493e-01 | -1.19905204e-01 | -1.52846128e-01 | -2.21726730e-01 |
| -2.94798970e-01 | -2.57434785e-01 | 5.40754311e-02  | -1.67930976e-03 |
| -4.73940261e-02 | -1.61136210e-01 | 8.02929252e-02  | -1.47165984e-01 |
| -8.32483247e-02 | -2.66365528e-01 | -1.61602870e-02 | 2.44220316e-01  |

|                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|
| 1.93133820e-02  | 4.64832708e-02  | -1.68969661e-01 | -1.35539040e-01 |
| 2.80833822e-02  | -1.82075635e-01 | -2.48233348e-01 | 3.43615860e-01  |
| 2.93125302e-01  | -7.08310306e-02 | 7.49630257e-02  | 1.97271705e-01  |
| 3.48101892e-02  | -3.97022702e-02 | 2.15118110e-01  | -4.50760908e-02 |
| -4.92423326e-02 | 4.58200090e-02  | 2.50818700e-01  | -3.72259915e-01 |
| 2.90479600e-01  | -1.36801824e-01 | 1.59229204e-01  | -1.13981202e-01 |
| -5.09829000e-02 | -2.31777996e-01 | 9.61414352e-02  | -2.37861909e-02 |
| -6.69691041e-02 | 3.93171102e-01  | -5.59290089e-02 | 3.28840345e-01  |
| -7.44685531e-02 | 3.27429511e-02  | -3.58317187e-03 | 1.25681549e-01  |
| -1.92001119e-01 | -1.84655458e-01 | -1.63177118e-01 | -1.60496421e-02 |
| 2.24943280e-01  | 5.18375859e-02  | -1.62278384e-01 | 2.45927334e-01  |
| -8.43128189e-02 | -3.00334334e-01 | 6.74607083e-02  | 1.31054908e-01  |
| 3.00829541e-02  | -4.17716831e-01 | 3.90796401e-02  | -1.55693606e-01 |
| -3.95431034e-02 | 4.80048060e-02  | -9.99328792e-02 | 1.96788222e-01  |
| 3.32303703e-01  | 3.19115235e-03  | 4.55405936e-02  | 1.98319420e-01  |
| -1.76160425e-01 | 3.68714154e-01  | 2.20427960e-02  | -6.78398311e-02 |
| -1.56519920e-01 | 1.39358282e-01  | 1.20546088e-01  | -5.60964979e-02 |
| -2.23587364e-01 | -2.09787577e-01 | -7.26886019e-02 | 1.96836904e-01  |
| 1.54818386e-01  | 2.06712037e-02  | -7.34864548e-02 | 4.29008901e-02  |
| -6.56196848e-02 | -2.08997920e-01 | 1.52214676e-01  | 1.60167709e-01  |
| 1.73863962e-01  | 3.96528840e-02  | -3.41452241e-01 | 4.36904468e-02  |
| 2.55822688e-01  | -2.95258850e-01 | 3.59045386e-01  | 1.27803208e-02  |
| 7.75438547e-02  | 2.68497407e-01  | 1.63680434e-01  | 2.46132612e-01  |
| -2.43196204e-01 | 1.54830664e-01  | 8.61508772e-03  | -1.82837114e-01 |
| 9.37864388e-06  | 7.14296475e-02  | -1.91982478e-01 | -1.51720911e-01 |
| -3.25180590e-02 | 6.15220308e-01  | 7.95181096e-02  | 1.46865115e-01  |
| 4.48604934e-02  | -4.82436985e-01 | -2.08653480e-01 | 6.52298611e-03  |
| -1.78304706e-02 | -1.34491175e-01 | 6.15640692e-02  | -4.17915955e-02 |
| 3.05843875e-02  | 4.12281275e-01  | 3.09576392e-01  | -1.18982010e-01 |
| -1.36158511e-01 | -1.43017560e-01 | -4.72919494e-01 | -5.96539192e-02 |
| -1.14213362e-01 | 1.76511899e-01  | -1.18436754e-01 | 3.82133164e-02  |
| -3.96532118e-02 | -3.80604297e-01 | -3.77102196e-03 | 7.64397308e-02  |
| 1.54263034e-01  | -1.73817858e-01 | 2.00220108e-01  | -1.39233083e-01 |
| 1.37678996e-01  | 7.24567920e-02  | 2.51273185e-01  | 1.07995719e-01  |
| -1.54602434e-02 | -3.32468841e-03 | -7.46456627e-03 | 2.63024986e-01  |
| -2.54204154e-01 | -1.06711388e-01 | 7.96224996e-02  | -4.24019188e-01 |
| -9.42714810e-02 | -1.19716860e-01 | 2.16964990e-01  | 1.13121115e-01  |
| 6.69243038e-01  | -2.80021280e-01 | -8.96213353e-02 | -9.37954858e-02 |
| -3.94240260e-01 | -5.21486923e-02 | -1.70960188e-01 | 2.23935664e-01  |
| -2.63201743e-01 | -2.14583501e-01 | 6.46317154e-02  | 2.28315175e-01  |
| 1.28163129e-01  | 1.95355177e-01  | -5.45718133e-01 | -1.04688955e-02 |
| 1.77491963e-01  | 2.67759487e-02  | -5.01418531e-01 | -1.34113491e-01 |
| -7.96580315e-02 | 4.10363544e-03  | -8.22813064e-03 | -4.31562185e-01 |
| -3.39084595e-01 | 1.42055424e-02  | -2.25144073e-01 | -1.87770441e-01 |
| 8.52994248e-02  | 9.91251841e-02  | -2.14395434e-01 | 1.10061809e-01  |
| -1.00015461e-01 | -2.49436736e-01 | 2.78937835e-02  | -1.94305666e-02 |
| -1.09906029e-02 | -3.52871835e-01 | 1.13414675e-01  | -5.05628660e-02 |
| 1.14488237e-01  | -6.74359977e-01 | -4.81343389e-01 | -3.43015581e-01 |
| 2.69350022e-01  | 1.21465370e-01  | -1.05356090e-02 | -2.71666586e-01 |
| -8.42319801e-02 | 3.39104205e-01  | 6.61544129e-02  | 1.96260929e-01  |
| -3.63996744e-01 | -6.58216923e-02 | -5.19641899e-02 | 3.80869061e-02  |
| 3.91338259e-01  | 3.04440141e-01  | -1.26176849e-01 | 1.56743288e-01  |
| 1.15540899e-01  | 2.57954448e-01  | -1.00360159e-02 | -2.29438215e-01 |
| -1.25840411e-01 | -6.74225911e-02 | -1.48194999e-01 | 9.18829069e-02  |
| -9.87157077e-02 | -1.99650154e-01 | 5.74524738e-02  | -1.23697296e-01 |
| -4.20444876e-01 | -1.25088841e-01 | -1.31912217e-01 | -1.35411844e-01 |
| -3.77414152e-02 | -2.38851830e-01 | -2.02091411e-01 | -1.08332280e-02 |

```
4.30721156e-02 -2.17161607e-02 2.25265279e-01 2.33677626e-01
-5.91066442e-02 -7.41588995e-02 -1.00842252e-01 4.72197622e-01
-2.51980305e-01 1.51866853e-01 3.33226740e-01 -5.22717349e-02
7.62687102e-02 -1.75606862e-01 2.28440086e-03 -2.05458939e-01
-2.37857014e-01 9.45121646e-02 -2.10536763e-01 1.83730870e-01
3.11037838e-01 -9.33999568e-02 3.01737357e-02 3.78455848e-01
9.30074081e-02 -1.07476041e-01 -3.05300683e-01 -6.87307939e-02] ->
[[0. 0. 0. 0. 0. 0. 1. 0. 0.]]
[-8.19788314e-03 1.93318456e-01 -2.55737096e-01 -5.85262999e-02
-1.12143733e-01 1.28226191e-01 -1.63523600e-01 3.29153985e-01
1.72688179e-02 -7.23662078e-02 -4.75730926e-01 -1.56640321e-01
-3.25731456e-01 6.37583286e-02 -3.48807164e-02 7.54046515e-02
-2.67262399e-01 -2.07818210e-01 -4.25361954e-02 1.84763186e-02
-2.99307972e-01 1.15987822e-01 -1.50392756e-01 -8.54285732e-02
-1.55638503e-02 -1.67371407e-01 5.10390336e-03 9.89027023e-02
-2.01399982e-01 3.10537755e-01 1.34452224e-01 1.30464569e-01
1.95780411e-01 5.01945540e-02 -5.31948149e-01 -1.85767025e-01
1.22942381e-01 7.54639953e-02 -1.22166574e-01 -4.99696508e-02
-6.83552548e-02 -6.58104420e-01 6.36971444e-02 1.41285956e-01
-1.44623920e-01 -8.98232311e-03 -3.38681191e-01 3.19188088e-01
3.95171702e-01 -5.40090322e-01 -5.58430795e-04 7.72019476e-02
3.45504552e-01 2.31432348e-01 5.54515064e-01 -3.62869292e-01
-1.29860565e-01 5.47985673e-01 6.30740598e-02 8.19746926e-02
-1.09874740e-01 2.18467295e-01 -3.48884016e-02 2.04096988e-01
-4.61730473e-02 1.16994128e-01 -1.67139530e-01 -9.06317532e-02
1.63974106e-01 -2.07277089e-01 -9.52801108e-02 -1.60821632e-01
1.71140358e-01 1.60285488e-01 2.45889872e-01 6.92768022e-02
-7.31295496e-02 2.27498814e-01 1.54240772e-01 -3.36584568e-01
2.05547854e-01 9.88019481e-02 7.80080399e-03 2.23548993e-01
3.24082553e-01 2.67415315e-01 -4.23092872e-01 -1.86155140e-01
-5.90210967e-02 1.89443082e-01 -2.48147652e-01 1.30945921e-03
7.92904645e-02 -9.25080553e-02 2.97049358e-02 1.24130942e-01
-4.43422854e-01 3.02665323e-01 -2.96746910e-01 2.79361755e-01
-1.00608163e-01 3.18943679e-01 2.59121776e-01 2.07691684e-01
-3.09031337e-01 2.35513002e-01 -4.21084911e-02 -2.48895317e-01
-9.59007740e-02 -3.03914875e-01 1.31026596e-01 3.34041193e-02
-4.23344195e-01 2.01583043e-01 2.56819606e-01 1.58521101e-01
9.14690793e-02 2.74726987e-01 -4.80642915e-02 -2.05094770e-01
4.60502543e-02 5.64997410e-03 6.29949346e-02 -1.33641645e-01
-1.22518921e-02 -2.19497427e-01 -4.34247665e-02 2.12650120e-01
-2.38290668e-01 1.47572299e-02 2.08088800e-01 1.95156559e-01
2.76555806e-01 -2.51404196e-02 -1.40751600e-01 1.54192463e-01
-1.01445548e-01 6.77790865e-02 1.42202742e-04 -3.94800723e-01
-2.78166771e-01 -2.08751321e-01 4.67118472e-01 6.68946281e-02
-1.24982662e-01 5.39936900e-01 2.07879215e-01 -6.27942264e-01
-5.96871190e-02 2.15298489e-01 -2.51414299e-01 8.56768340e-02
-1.22809909e-01 3.64094041e-02 -1.87768579e-01 6.21910393e-02
-8.15605223e-02 1.15067698e-01 2.27230117e-02 -7.60399923e-02
9.59183797e-02 -2.26689413e-01 -3.21219891e-01 2.80338734e-01
-3.12000751e-01 7.85248056e-02 -4.90650646e-02 -1.91503912e-01
-2.25585863e-01 -3.92571986e-01 -1.40992152e-02 -1.17863886e-01
8.60270932e-02 7.31864152e-03 4.06437427e-01 -1.05658554e-01
-3.58285278e-01 3.86391789e-01 -8.41783639e-03 -1.05645679e-01
-6.26050457e-02 -3.01659763e-01 1.03028394e-01 5.19222438e-01
1.95978776e-01 -1.01899147e-01 3.13420415e-01 5.11938557e-02
2.95158833e-01 -3.36767063e-02 4.48371470e-02 5.21192193e-01
2.78637022e-01 -4.09705877e-01 3.65453474e-02 2.09444687e-01
```

|                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|
| -2.35680997e-01 | -5.46684444e-01 | -6.87311217e-03 | 2.94725299e-01  |
| -5.34977615e-01 | -1.52939141e-01 | 1.93228826e-01  | 2.54261613e-01  |
| 1.77904129e-01  | 1.94128066e-01  | -4.10632163e-01 | 3.25125724e-01  |
| 2.31717136e-02  | -4.24731374e-01 | -5.85962355e-01 | 2.19214670e-02  |
| -2.25591764e-01 | -2.19828025e-01 | -2.28205118e-02 | -3.72632384e-01 |
| -2.67896384e-01 | -1.27756312e-01 | 1.93230763e-01  | -2.01365992e-01 |
| 5.95798641e-02  | 2.77440786e-01  | 8.24818164e-02  | -1.47691175e-01 |
| -5.18139005e-01 | -2.19808012e-01 | 1.38957992e-01  | 1.38900071e-01  |
| -9.44016427e-02 | -2.24219382e-01 | 2.07139924e-01  | -2.33354717e-01 |
| -5.80870919e-02 | -1.57668680e-01 | -7.70893544e-02 | 7.81492069e-02  |
| -6.59244657e-02 | 1.69186592e-01  | -1.44687667e-01 | -2.11417049e-01 |
| -7.01892823e-02 | -3.96145061e-02 | -5.46510458e-01 | 1.28850058e-01  |
| 2.22768903e-01  | -2.98677236e-01 | -1.18267253e-01 | 6.33210540e-02  |
| -5.74567914e-02 | -1.67560130e-01 | -1.39726549e-01 | 2.93317251e-02  |
| 1.06791906e-01  | -5.84209077e-02 | 6.79801702e-02  | -2.59042174e-01 |
| -2.82391757e-01 | -2.19578519e-01 | -3.81942928e-01 | 2.21975192e-01  |
| 3.05138379e-02  | 1.92736208e-01  | -1.85888261e-01 | -1.06375091e-01 |
| -2.32108030e-02 | -6.32791370e-02 | -1.58603340e-01 | 9.82918311e-03  |
| -1.30309612e-01 | -1.20525494e-01 | -4.22838628e-01 | -2.88072854e-01 |
| -5.22131622e-02 | -4.51118201e-02 | 2.37653092e-01  | 6.80113658e-02  |
| -5.76707199e-02 | -4.53665197e-01 | 1.98694944e-01  | 6.51667640e-02  |
| -3.39328945e-01 | 3.13593984e-01  | 8.70105624e-02  | -1.60077110e-01 |
| -1.27475768e-01 | -4.41596538e-01 | 7.60736838e-02  | -1.28953993e-01 |
| -3.30101579e-01 | -2.48410963e-02 | -2.69903317e-02 | -3.11167035e-02 |
| -1.59395128e-01 | 1.90406851e-02  | -1.47860155e-01 | 7.09712058e-02  |
| 5.60208082e-01  | 1.01584129e-01  | -3.00363451e-01 | -1.45654321e-01 |
| -1.28002688e-01 | -1.52486414e-01 | -3.12842727e-01 | 4.64192731e-03  |
| -2.96131939e-01 | -1.98212892e-01 | -1.51366189e-01 | 2.97824711e-01  |
| -2.23264307e-01 | 8.09413791e-02  | -2.10164845e-01 | -1.84343770e-01 |
| -1.78107738e-01 | -1.64515167e-01 | 4.20021296e-01  | 5.25286682e-02  |
| 2.76020393e-02  | 1.17821638e-02  | -9.70714539e-02 | -6.89446330e-02 |
| 2.81050026e-01  | 1.86055824e-02  | 1.15677491e-01  | 9.43794497e-04  |
| -1.11368045e-01 | -1.97569832e-01 | -1.38451830e-01 | 5.19487299e-02  |
| -2.94302493e-01 | -1.19905204e-01 | -1.52846128e-01 | -2.21726730e-01 |
| -2.94798970e-01 | -2.57434785e-01 | 5.40754311e-02  | -1.67930976e-03 |
| -4.73940261e-02 | -1.61136210e-01 | 8.02929252e-02  | -1.47165984e-01 |
| -8.32483247e-02 | -2.66365528e-01 | -1.61602870e-02 | 2.44220316e-01  |
| 1.93133820e-02  | 4.64832708e-02  | -1.68969661e-01 | -1.35539040e-01 |
| 2.80833822e-02  | -1.82075635e-01 | -2.48233348e-01 | 3.43615860e-01  |
| 2.93125302e-01  | -7.08310306e-02 | 7.49630257e-02  | 1.97271705e-01  |
| 3.48101892e-02  | -3.97022702e-02 | 2.15118110e-01  | -4.50760908e-02 |
| -4.92423326e-02 | 4.58200090e-02  | 2.50818700e-01  | -3.72259915e-01 |
| 2.90479600e-01  | -1.36801824e-01 | 1.59229204e-01  | -1.13981202e-01 |
| -5.09829000e-02 | -2.31777996e-01 | 9.61414352e-02  | -2.37861909e-02 |
| -6.69691041e-02 | 3.93171102e-01  | -5.59290089e-02 | 3.28840345e-01  |
| -7.44685531e-02 | 3.27429511e-02  | -3.58317187e-03 | 1.25681549e-01  |
| -1.92001119e-01 | -1.84655458e-01 | -1.63177118e-01 | -1.60496421e-02 |
| 2.24943280e-01  | 5.18375859e-02  | -1.62278384e-01 | 2.45927334e-01  |
| -8.43128189e-02 | -3.00334334e-01 | 6.74607083e-02  | 1.31054908e-01  |
| 3.00829541e-02  | -4.17716831e-01 | 3.90796401e-02  | -1.55693606e-01 |
| -3.95431034e-02 | 4.80048060e-02  | -9.99328792e-02 | 1.96788222e-01  |
| 3.32303703e-01  | 3.19115235e-03  | 4.55405936e-02  | 1.98319420e-01  |
| -1.76160425e-01 | 3.68714154e-01  | 2.20427960e-02  | -6.78398311e-02 |
| -1.56519920e-01 | 1.39358282e-01  | 1.20546088e-01  | -5.60964979e-02 |
| -2.23587364e-01 | -2.09787577e-01 | -7.26886019e-02 | 1.96836904e-01  |
| 1.54818386e-01  | 2.06712037e-02  | -7.34864548e-02 | 4.29008901e-02  |
| -6.56196848e-02 | -2.08997920e-01 | 1.52214676e-01  | 1.60167709e-01  |

|                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|
| 1.73863962e-01  | 3.96528840e-02  | -3.41452241e-01 | 4.36904468e-02  |
| 2.55822688e-01  | -2.95258850e-01 | 3.59045386e-01  | 1.27803208e-02  |
| 7.75438547e-02  | 2.68497407e-01  | 1.63680434e-01  | 2.46132612e-01  |
| -2.43196204e-01 | 1.54830664e-01  | 8.61508772e-03  | -1.82837114e-01 |
| 9.37864388e-06  | 7.14296475e-02  | -1.91982478e-01 | -1.51720911e-01 |
| -3.25180590e-02 | 6.15220308e-01  | 7.95181096e-02  | 1.46865115e-01  |
| 4.48604934e-02  | -4.82436985e-01 | -2.08653480e-01 | 6.52298611e-03  |
| -1.78304706e-02 | -1.34491175e-01 | 6.15640692e-02  | -4.17915955e-02 |
| 3.05843875e-02  | 4.12281275e-01  | 3.09576392e-01  | -1.18982010e-01 |
| -1.36158511e-01 | -1.43017560e-01 | -4.72919494e-01 | -5.96539192e-02 |
| -1.14213362e-01 | 1.76511899e-01  | -1.18436754e-01 | 3.82133164e-02  |
| -3.96532118e-02 | -3.80604297e-01 | -3.77102196e-03 | 7.64397308e-02  |
| 1.54263034e-01  | -1.73817858e-01 | 2.00220108e-01  | -1.39233083e-01 |
| 1.37678996e-01  | 7.24567920e-02  | 2.51273185e-01  | 1.07995719e-01  |
| -1.54602434e-02 | -3.32468841e-03 | -7.46456627e-03 | 2.63024986e-01  |
| -2.54204154e-01 | -1.06711388e-01 | 7.96224996e-02  | -4.24019188e-01 |
| -9.42714810e-02 | -1.19716860e-01 | 2.16964990e-01  | 1.13121115e-01  |
| 6.69243038e-01  | -2.80021280e-01 | -8.96213353e-02 | -9.37954858e-02 |
| -3.94240260e-01 | -5.21486923e-02 | -1.70960188e-01 | 2.23935664e-01  |
| -2.63201743e-01 | -2.14583501e-01 | 6.46317154e-02  | 2.28315175e-01  |
| 1.28163129e-01  | 1.95355177e-01  | -5.45718133e-01 | -1.04688955e-02 |
| 1.77491963e-01  | 2.67759487e-02  | -5.01418531e-01 | -1.34113491e-01 |
| -7.96580315e-02 | 4.10363544e-03  | -8.22813064e-03 | -4.31562185e-01 |
| -3.39084595e-01 | 1.42055424e-02  | -2.25144073e-01 | -1.87770441e-01 |
| 8.52994248e-02  | 9.91251841e-02  | -2.14395434e-01 | 1.10061809e-01  |
| -1.00015461e-01 | -2.49436736e-01 | 2.78937835e-02  | -1.94305666e-02 |
| -1.09906029e-02 | -3.52871835e-01 | 1.13414675e-01  | -5.05628660e-02 |
| 1.14488237e-01  | -6.74359977e-01 | -4.81343389e-01 | -3.43015581e-01 |
| 2.69350022e-01  | 1.21465370e-01  | -1.05356090e-02 | -2.71666586e-01 |
| -8.42319801e-02 | 3.39104205e-01  | 6.61544129e-02  | 1.96260929e-01  |
| -3.63996744e-01 | -6.58216923e-02 | -5.19641899e-02 | 3.80869061e-02  |
| 3.91338259e-01  | 3.04440141e-01  | -1.26176849e-01 | 1.56743288e-01  |
| 1.15540899e-01  | 2.57954448e-01  | -1.00360159e-02 | -2.29438215e-01 |
| -1.25840411e-01 | -6.74225911e-02 | -1.48194999e-01 | 9.18829069e-02  |
| -9.87157077e-02 | -1.99650154e-01 | 5.74524738e-02  | -1.23697296e-01 |
| -4.20444876e-01 | -1.25088841e-01 | -1.31912217e-01 | -1.35411844e-01 |
| -3.77414152e-02 | -2.38851830e-01 | -2.02091411e-01 | -1.08332280e-02 |
| 4.30721156e-02  | -2.17161607e-02 | 2.25265279e-01  | 2.33677626e-01  |
| -5.91066442e-02 | -7.41588995e-02 | -1.00842252e-01 | 4.72197622e-01  |
| -2.51980305e-01 | 1.51866853e-01  | 3.33226740e-01  | -5.22717349e-02 |
| 7.62687102e-02  | -1.75606862e-01 | 2.28440086e-03  | -2.05458939e-01 |
| -2.37857014e-01 | 9.45121646e-02  | -2.10536763e-01 | 1.83730870e-01  |
| 3.11037838e-01  | -9.33999568e-02 | 3.01737357e-02  | 3.78455848e-01  |
| 9.30074081e-02  | -1.07476041e-01 | -3.05300683e-01 | -6.87307939e-02 |
| 1.59357086e-01  | 1.18073292e-01  | -1.94542333e-01 | 1.72432184e-01  |
| -7.51207471e-02 | 8.31844583e-02  | 2.93652594e-01  | 2.38601550e-01  |
| 3.76580050e-03  | -6.98884949e-02 | -9.80963930e-02 | 5.26036099e-02  |
| -2.58915722e-01 | -2.89399177e-02 | -5.58336601e-02 | 2.19740853e-01  |
| -2.85466790e-01 | 8.44207928e-02  | -3.61072905e-02 | 1.41626269e-01  |
| -1.42787814e-01 | 3.76672328e-01  | -2.32369769e-02 | -2.08301976e-01 |
| -1.52429715e-01 | -7.83066731e-03 | -2.94688255e-01 | 2.94639796e-01  |
| 1.82755157e-01  | 9.84446481e-02  | 4.94708158e-02  | -7.93528333e-02 |
| -9.37881842e-02 | -4.23773170e-01 | -1.28640324e-01 | 7.58199245e-02  |
| -1.46779835e-01 | 1.69399544e-03  | -2.23073792e-02 | -1.73199147e-01 |
| -2.59192646e-01 | -3.58933747e-01 | -1.37308046e-01 | 2.60157824e-01  |
| -1.97335675e-01 | -1.55454293e-01 | 8.69532898e-02  | -4.95323092e-02 |
| 3.45346361e-01  | -1.05545014e-01 | -1.81371689e-01 | 1.04653984e-01  |



|                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|
| 2.13711470e-01  | 1.54687926e-01  | 2.54383266e-01  | 2.62094080e-01  |
| -2.61373967e-01 | -1.23798158e-02 | -1.26662711e-02 | 6.88340217e-02  |
| -1.64214492e-01 | -2.12840840e-01 | 2.21722767e-01  | 8.62467661e-02  |
| 2.74025261e-01  | -1.65959969e-01 | 3.86081755e-01  | 1.35656148e-01  |
| 2.16811106e-01  | 9.85317603e-02  | -9.11876038e-02 | -1.60227716e-01 |
| 2.80423045e-01  | -2.39790920e-02 | 1.67653173e-01  | 2.71800995e-01  |
| 6.17754348e-02  | 2.24040598e-02  | 9.22093093e-02  | 3.58554982e-02  |
| 5.03476113e-02  | -6.10370524e-02 | -9.56719816e-02 | -2.19227239e-01 |
| -5.83785921e-02 | 1.19145699e-01  | -6.15942888e-02 | 2.45511085e-02  |
| -3.01883426e-02 | -8.48109201e-02 | -6.05366640e-02 | -2.96960562e-01 |
| 1.31319687e-01  | -4.40078765e-01 | 3.71894203e-02  | 1.01576880e-01  |
| 5.75591624e-02  | 5.72020561e-02  | 2.45402548e-02  | 3.02145660e-01  |
| 1.42727375e-01  | 1.44829214e-01  | 2.60785788e-01  | -5.58878295e-03 |
| -1.93750709e-01 | 8.86656791e-02  | 3.59328054e-02  | -4.29539025e-01 |
| -2.73156147e-02 | 1.95869148e-01  | -2.36021504e-01 | 2.08259560e-02  |
| -1.87755510e-01 | -6.31835908e-02 | -4.70160067e-01 | 1.75316438e-01  |
| 7.12417811e-02  | 1.83751751e-02  | -7.45476261e-02 | -5.40569499e-02 |
| -1.52935624e-01 | 1.74953878e-01  | 1.91645533e-01  | 1.90425202e-01  |
| 3.96441445e-02  | 3.73551063e-02  | 1.11061566e-01  | 3.26659307e-02  |
| 8.74631628e-02  | -5.50932229e-01 | 3.43769863e-02  | 2.63986886e-02  |
| 4.21343625e-01  | 1.47270188e-01  | 8.56536180e-02  | -9.92612168e-02 |
| 1.44661382e-01  | 3.14606696e-01  | 2.71684110e-01  | 1.26298666e-01  |
| 3.06872148e-02  | 3.48012485e-02  | 2.57874373e-02  | 1.17914649e-02  |
| -5.20234928e-02 | 4.34443444e-01  | -1.43969163e-01 | -2.34416232e-01 |
| -2.70714521e-01 | -3.52176949e-02 | -5.13002753e-01 | 2.77747601e-01  |
| 2.17147171e-01  | -5.23802228e-02 | -9.84058157e-02 | -2.99813986e-01 |
| -2.59716898e-01 | 3.00835632e-02  | -1.09366976e-01 | 6.21630922e-02  |
| -6.17096573e-02 | 3.29713225e-02  | -2.72397995e-02 | 8.30065757e-02  |
| 9.75671783e-02  | 2.69979239e-01  | 5.43040149e-02  | 2.74037361e-01  |
| 1.21071003e-03  | -1.12680174e-01 | -9.44447741e-02 | 1.42977042e-02  |
| 1.60134882e-02  | 1.47059709e-01  | 9.26311910e-02  | -2.42051169e-01 |
| -7.59890005e-02 | 8.07066411e-02  | 9.18080509e-02  | 3.84047665e-02  |
| -2.59923220e-01 | 1.51593471e-02  | -1.26153037e-01 | 2.80143797e-01  |
| 1.12403832e-01  | 3.45987380e-01  | 6.74737990e-02  | 1.70807511e-01  |
| -4.44618203e-02 | 6.17295578e-02  | 5.00851218e-03  | 1.12559021e-01  |
| 1.97814941e-01  | -1.19529001e-01 | -1.39373004e-01 | 1.62549466e-01  |
| 1.47074938e-01  | 7.06764758e-02  | 2.93673705e-02  | 2.09786609e-01  |
| -5.31781651e-02 | 9.17047411e-02  | 1.75700516e-01  | 1.67383745e-01  |
| 5.17717116e-02  | 1.02442734e-01  | -1.42310247e-01 | 3.32933545e-01  |
| 2.04754740e-01  | 7.37320585e-03  | -4.00519937e-01 | -2.37964109e-01 |
| 1.63770482e-01  | -9.49526653e-02 | -1.33547066e-02 | -2.27290720e-01 |
| -2.09703341e-01 | -1.28782749e-01 | -2.53570657e-02 | 6.88151494e-02  |
| 4.53894064e-02  | 1.90416068e-01  | 4.06187847e-02  | 6.29945919e-02  |
| 1.20325172e-02  | 3.60493772e-02  | 7.66044185e-02  | -2.46364936e-01 |
| 2.60074008e-02  | -1.37602091e-01 | -5.10821044e-02 | -2.10424602e-01 |
| 2.63544321e-01  | -2.73499757e-01 | -2.64665991e-01 | -1.44399717e-01 |
| 4.65553738e-02  | 6.31644055e-02  | -1.23298235e-01 | 8.65715221e-02  |
| -2.83900201e-01 | -1.05851255e-01 | -2.15419114e-01 | -7.30885118e-02 |
| 6.74026310e-02  | 1.31581053e-01  | -1.37944177e-01 | 3.49872231e-01  |
| -5.25776632e-02 | 4.02169898e-02  | -1.78974822e-01 | 5.65098673e-02  |
| 3.82037431e-01  | 1.52979568e-01  | -1.85733184e-01 | 1.65536225e-01  |
| 1.06043026e-01  | -1.44612983e-01 | -2.92519242e-01 | 9.68883559e-02  |
| -2.69471169e-01 | -3.11438460e-02 | -2.00801656e-01 | -3.56719159e-02 |
| 7.69059956e-02  | 2.08680294e-02  | -8.46385732e-02 | -2.77631015e-01 |
| 2.26994887e-01  | -1.02845930e-01 | 2.16934737e-02  | -1.67850420e-01 |
| 3.97305712e-02  | -1.15134278e-02 | 2.04697713e-01  | -1.89971268e-01 |
| -2.95838356e-01 | -2.76767105e-01 | 1.63810968e-01  | 8.37999806e-02  |

```
1.48658127e-01 -1.37286305e-01 1.35871202e-01 -1.23310111e-01
1.32110521e-01 -3.54248881e-02 1.44168869e-01 1.09475702e-01
-4.60087329e-01 1.54884130e-01 1.27283320e-01 2.58040756e-01
-4.75979187e-02 5.11781983e-02 -2.77065516e-01 -7.83998743e-02
2.54693747e-01 1.91429079e-01 2.72032797e-01 1.31090414e-02] ->
[[0. 0. 0. 0. 0. 0. 0. 1. 0.]]
[-1.28002688e-01 -1.52486414e-01 -3.12842727e-01 4.64192731e-03
-2.96131939e-01 -1.98212892e-01 -1.51366189e-01 2.97824711e-01
-2.23264307e-01 8.09413791e-02 -2.10164845e-01 -1.84343770e-01
-1.78107738e-01 -1.64515167e-01 4.20021296e-01 5.25286682e-02
2.76020393e-02 1.17821638e-02 -9.70714539e-02 -6.89446330e-02
2.81050026e-01 1.86055824e-02 1.15677491e-01 9.43794497e-04
-1.11368045e-01 -1.97569832e-01 -1.38451830e-01 5.19487299e-02
-2.94302493e-01 -1.19905204e-01 -1.52846128e-01 -2.21726730e-01
-2.94798970e-01 -2.57434785e-01 5.40754311e-02 -1.67930976e-03
-4.73940261e-02 -1.61136210e-01 8.02929252e-02 -1.47165984e-01
-8.32483247e-02 -2.66365528e-01 -1.61602870e-02 2.44220316e-01
1.93133820e-02 4.64832708e-02 -1.68969661e-01 -1.35539040e-01
2.80833822e-02 -1.82075635e-01 -2.48233348e-01 3.43615860e-01
2.93125302e-01 -7.08310306e-02 7.49630257e-02 1.97271705e-01
3.48101892e-02 -3.97022702e-02 2.15118110e-01 -4.50760908e-02
-4.92423326e-02 4.58200090e-02 2.50818700e-01 -3.72259915e-01
2.90479600e-01 -1.36801824e-01 1.59229204e-01 -1.13981202e-01
-5.09829000e-02 -2.31777996e-01 9.61414352e-02 -2.37861909e-02
-6.69691041e-02 3.93171102e-01 -5.59290089e-02 3.28840345e-01
-7.44685531e-02 3.27429511e-02 -3.58317187e-03 1.25681549e-01
-1.92001119e-01 -1.84655458e-01 -1.63177118e-01 -1.60496421e-02
2.24943280e-01 5.18375859e-02 -1.62278384e-01 2.45927334e-01
-8.43128189e-02 -3.00334334e-01 6.74607083e-02 1.31054908e-01
3.00829541e-02 -4.17716831e-01 3.90796401e-02 -1.55693606e-01
-3.95431034e-02 4.80048060e-02 -9.99328792e-02 1.96788222e-01
3.32303703e-01 3.19115235e-03 4.55405936e-02 1.98319420e-01
-1.76160425e-01 3.68714154e-01 2.20427960e-02 -6.78398311e-02
-1.56519920e-01 1.39358282e-01 1.20546088e-01 -5.60964979e-02
-2.23587364e-01 -2.09787577e-01 -7.26886019e-02 1.96836904e-01
1.54818386e-01 2.06712037e-02 -7.34864548e-02 4.29008901e-02
-6.56196848e-02 -2.08997920e-01 1.52214676e-01 1.60167709e-01
1.73863962e-01 3.96528840e-02 -3.41452241e-01 4.36904468e-02
2.55822688e-01 -2.95258850e-01 3.59045386e-01 1.27803208e-02
7.75438547e-02 2.68497407e-01 1.63680434e-01 2.46132612e-01
-2.43196204e-01 1.54830664e-01 8.61508772e-03 -1.82837114e-01
9.37864388e-06 7.14296475e-02 -1.91982478e-01 -1.51720911e-01
-3.25180590e-02 6.15220308e-01 7.95181096e-02 1.46865115e-01
4.48604934e-02 -4.82436985e-01 -2.08653480e-01 6.52298611e-03
-1.78304706e-02 -1.34491175e-01 6.15640692e-02 -4.17915955e-02
3.05843875e-02 4.12281275e-01 3.09576392e-01 -1.18982010e-01
-1.36158511e-01 -1.43017560e-01 -4.72919494e-01 -5.96539192e-02
-1.14213362e-01 1.76511899e-01 -1.18436754e-01 3.82133164e-02
-3.96532118e-02 -3.80604297e-01 -3.77102196e-03 7.64397308e-02
1.54263034e-01 -1.73817858e-01 2.00220108e-01 -1.39233083e-01
1.37678996e-01 7.24567920e-02 2.51273185e-01 1.07995719e-01
-1.54602434e-02 -3.32468841e-03 -7.46456627e-03 2.63024986e-01
-2.54204154e-01 -1.06711388e-01 7.96224996e-02 -4.24019188e-01
-9.42714810e-02 -1.19716860e-01 2.16964990e-01 1.13121115e-01
6.69243038e-01 -2.80021280e-01 -8.96213353e-02 -9.37954858e-02
-3.94240260e-01 -5.21486923e-02 -1.70960188e-01 2.23935664e-01
-2.63201743e-01 -2.14583501e-01 6.46317154e-02 2.28315175e-01
```

|                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|
| 1.28163129e-01  | 1.95355177e-01  | -5.45718133e-01 | -1.04688955e-02 |
| 1.77491963e-01  | 2.67759487e-02  | -5.01418531e-01 | -1.34113491e-01 |
| -7.96580315e-02 | 4.10363544e-03  | -8.22813064e-03 | -4.31562185e-01 |
| -3.39084595e-01 | 1.42055424e-02  | -2.25144073e-01 | -1.87770441e-01 |
| 8.52994248e-02  | 9.91251841e-02  | -2.14395434e-01 | 1.10061809e-01  |
| -1.00015461e-01 | -2.49436736e-01 | 2.78937835e-02  | -1.94305666e-02 |
| -1.09906029e-02 | -3.52871835e-01 | 1.13414675e-01  | -5.05628660e-02 |
| 1.14488237e-01  | -6.74359977e-01 | -4.81343389e-01 | -3.43015581e-01 |
| 2.69350022e-01  | 1.21465370e-01  | -1.05356090e-02 | -2.71666586e-01 |
| -8.42319801e-02 | 3.39104205e-01  | 6.61544129e-02  | 1.96260929e-01  |
| -3.63996744e-01 | -6.58216923e-02 | -5.19641899e-02 | 3.80869061e-02  |
| 3.91338259e-01  | 3.04440141e-01  | -1.26176849e-01 | 1.56743288e-01  |
| 1.15540899e-01  | 2.57954448e-01  | -1.00360159e-02 | -2.29438215e-01 |
| -1.25840411e-01 | -6.74225911e-02 | -1.48194999e-01 | 9.18829069e-02  |
| -9.87157077e-02 | -1.99650154e-01 | 5.74524738e-02  | -1.23697296e-01 |
| -4.20444876e-01 | -1.25088841e-01 | -1.31912217e-01 | -1.35411844e-01 |
| -3.77414152e-02 | -2.38851830e-01 | -2.02091411e-01 | -1.08332280e-02 |
| 4.30721156e-02  | -2.17161607e-02 | 2.25265279e-01  | 2.33677626e-01  |
| -5.91066442e-02 | -7.41588995e-02 | -1.00842252e-01 | 4.72197622e-01  |
| -2.51980305e-01 | 1.51866853e-01  | 3.33226740e-01  | -5.22717349e-02 |
| 7.62687102e-02  | -1.75606862e-01 | 2.28440086e-03  | -2.05458939e-01 |
| -2.37857014e-01 | 9.45121646e-02  | -2.10536763e-01 | 1.83730870e-01  |
| 3.11037838e-01  | -9.33999568e-02 | 3.01737357e-02  | 3.78455848e-01  |
| 9.30074081e-02  | -1.07476041e-01 | -3.05300683e-01 | -6.87307939e-02 |
| 1.59357086e-01  | 1.18073292e-01  | -1.94542333e-01 | 1.72432184e-01  |
| -7.51207471e-02 | 8.31844583e-02  | 2.93652594e-01  | 2.38601550e-01  |
| 3.76580050e-03  | -6.98884949e-02 | -9.80963930e-02 | 5.26036099e-02  |
| -2.58915722e-01 | -2.89399177e-02 | -5.58336601e-02 | 2.19740853e-01  |
| -2.85466790e-01 | 8.44207928e-02  | -3.61072905e-02 | 1.41626269e-01  |
| -1.42787814e-01 | 3.76672328e-01  | -2.32369769e-02 | -2.08301976e-01 |
| -1.52429715e-01 | -7.83066731e-03 | -2.94688255e-01 | 2.94639796e-01  |
| 1.82755157e-01  | 9.84446481e-02  | 4.94708158e-02  | -7.93528333e-02 |
| -9.37881842e-02 | -4.23773170e-01 | -1.28640324e-01 | 7.58199245e-02  |
| -1.46779835e-01 | 1.69399544e-03  | -2.23073792e-02 | -1.73199147e-01 |
| -2.59192646e-01 | -3.58933747e-01 | -1.37308046e-01 | 2.60157824e-01  |
| -1.97335675e-01 | -1.55454293e-01 | 8.69532898e-02  | -4.95323092e-02 |
| 3.45346361e-01  | -1.05545014e-01 | -1.81371689e-01 | 1.04653984e-01  |
| 2.13711470e-01  | 1.54687926e-01  | 2.54383266e-01  | 2.62094080e-01  |
| -2.61373967e-01 | -1.23798158e-02 | -1.26662711e-02 | 6.88340217e-02  |
| -1.64214492e-01 | -2.12840840e-01 | 2.21722767e-01  | 8.62467661e-02  |
| 2.74025261e-01  | -1.65959969e-01 | 3.86081755e-01  | 1.35656148e-01  |
| 2.16811106e-01  | 9.85317603e-02  | -9.11876038e-02 | -1.60227716e-01 |
| 2.80423045e-01  | -2.39790920e-02 | 1.67653173e-01  | 2.71800995e-01  |
| 6.17754348e-02  | 2.24040598e-02  | 9.22093093e-02  | 3.58554982e-02  |
| 5.03476113e-02  | -6.10370524e-02 | -9.56719816e-02 | -2.19227239e-01 |
| -5.83785921e-02 | 1.19145699e-01  | -6.15942888e-02 | 2.45511085e-02  |
| -3.01883426e-02 | -8.48109201e-02 | -6.05366640e-02 | -2.96960562e-01 |
| 1.31319687e-01  | -4.40078765e-01 | 3.71894203e-02  | 1.01576880e-01  |
| 5.75591624e-02  | 5.72020561e-02  | 2.45402548e-02  | 3.02145660e-01  |
| 1.42727375e-01  | 1.44829214e-01  | 2.60785788e-01  | -5.58878295e-03 |
| -1.93750709e-01 | 8.86656791e-02  | 3.59328054e-02  | -4.29539025e-01 |
| -2.73156147e-02 | 1.95869148e-01  | -2.36021504e-01 | 2.08259560e-02  |
| -1.87755510e-01 | -6.31835908e-02 | -4.70160067e-01 | 1.75316438e-01  |
| 7.12417811e-02  | 1.83751751e-02  | -7.45476261e-02 | -5.40569499e-02 |
| -1.52935624e-01 | 1.74953878e-01  | 1.91645533e-01  | 1.90425202e-01  |
| 3.96441445e-02  | 3.73551063e-02  | 1.11061566e-01  | 3.26659307e-02  |
| 8.74631628e-02  | -5.50932229e-01 | 3.43769863e-02  | 2.63986886e-02  |

|                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|
| 4.21343625e-01  | 1.47270188e-01  | 8.56536180e-02  | -9.92612168e-02 |
| 1.44661382e-01  | 3.14606696e-01  | 2.71684110e-01  | 1.26298666e-01  |
| 3.06872148e-02  | 3.48012485e-02  | 2.57874373e-02  | 1.17914649e-02  |
| -5.20234928e-02 | 4.34443444e-01  | -1.43969163e-01 | -2.34416232e-01 |
| -2.70714521e-01 | -3.52176949e-02 | -5.13002753e-01 | 2.77747601e-01  |
| 2.17147171e-01  | -5.23802228e-02 | -9.84058157e-02 | -2.99813986e-01 |
| -2.59716898e-01 | 3.00835632e-02  | -1.09366976e-01 | 6.21630922e-02  |
| -6.17096573e-02 | 3.29713225e-02  | -2.72397995e-02 | 8.30065757e-02  |
| 9.75671783e-02  | 2.69979239e-01  | 5.43040149e-02  | 2.74037361e-01  |
| 1.21071003e-03  | -1.12680174e-01 | -9.44447741e-02 | 1.42977042e-02  |
| 1.60134882e-02  | 1.47059709e-01  | 9.26311910e-02  | -2.42051169e-01 |
| -7.59890005e-02 | 8.07066411e-02  | 9.18080509e-02  | 3.84047665e-02  |
| -2.59923220e-01 | 1.51593471e-02  | -1.26153037e-01 | 2.80143797e-01  |
| 1.12403832e-01  | 3.45987380e-01  | 6.74737990e-02  | 1.70807511e-01  |
| -4.44618203e-02 | 6.17295578e-02  | 5.00851218e-03  | 1.12559021e-01  |
| 1.97814941e-01  | -1.19529001e-01 | -1.39373004e-01 | 1.62549466e-01  |
| 1.47074938e-01  | 7.06764758e-02  | 2.93673705e-02  | 2.09786609e-01  |
| -5.31781651e-02 | 9.17047411e-02  | 1.75700516e-01  | 1.67383745e-01  |
| 5.17717116e-02  | 1.02442734e-01  | -1.42310247e-01 | 3.32933545e-01  |
| 2.04754740e-01  | 7.37320585e-03  | -4.00519937e-01 | -2.37964109e-01 |
| 1.63770482e-01  | -9.49526653e-02 | -1.33547066e-02 | -2.27290720e-01 |
| -2.09703341e-01 | -1.28782749e-01 | -2.53570657e-02 | 6.88151494e-02  |
| 4.53894064e-02  | 1.90416068e-01  | 4.06187847e-02  | 6.29945919e-02  |
| 1.20325172e-02  | 3.60493772e-02  | 7.66044185e-02  | -2.46364936e-01 |
| 2.60074008e-02  | -1.37602091e-01 | -5.10821044e-02 | -2.10424602e-01 |
| 2.63544321e-01  | -2.73499757e-01 | -2.64665991e-01 | -1.44399717e-01 |
| 4.65553738e-02  | 6.31644055e-02  | -1.23298235e-01 | 8.65715221e-02  |
| -2.83900201e-01 | -1.05851255e-01 | -2.15419114e-01 | -7.30885118e-02 |
| 6.74026310e-02  | 1.31581053e-01  | -1.37944177e-01 | 3.49872231e-01  |
| -5.25776632e-02 | 4.02169898e-02  | -1.78974822e-01 | 5.65098673e-02  |
| 3.82037431e-01  | 1.52979568e-01  | -1.85733184e-01 | 1.65536225e-01  |
| 1.06043026e-01  | -1.44612983e-01 | -2.92519242e-01 | 9.68883559e-02  |
| -2.69471169e-01 | -3.11438460e-02 | -2.00801656e-01 | -3.56719159e-02 |
| 7.69059956e-02  | 2.08680294e-02  | -8.46385732e-02 | -2.77631015e-01 |
| 2.26994887e-01  | -1.02845930e-01 | 2.16934737e-02  | -1.67850420e-01 |
| 3.97305712e-02  | -1.15134278e-02 | 2.04697713e-01  | -1.89971268e-01 |
| -2.95838356e-01 | -2.76767105e-01 | 1.63810968e-01  | 8.37999806e-02  |
| 1.48658127e-01  | -1.37286305e-01 | 1.35871202e-01  | -1.23310111e-01 |
| 1.32110521e-01  | -3.54248881e-02 | 1.44168869e-01  | 1.09475702e-01  |
| -4.60087329e-01 | 1.54884130e-01  | 1.27283320e-01  | 2.58040756e-01  |
| -4.75979187e-02 | 5.11781983e-02  | -2.77065516e-01 | -7.83998743e-02 |
| 2.54693747e-01  | 1.91429079e-01  | 2.72032797e-01  | 1.31090414e-02  |
| 2.00696941e-02  | 2.17040144e-02  | -2.73649812e-01 | -1.45588383e-01 |
| -3.86551917e-02 | -8.92118141e-02 | 6.72461763e-02  | 4.33792174e-02  |
| -1.62340939e-01 | -1.50650561e-01 | -6.45114705e-02 | -2.46862292e-01 |
| -4.02667105e-01 | 5.15187196e-02  | 8.27926174e-02  | 1.42826930e-01  |
| -2.50022411e-01 | -2.28285328e-01 | 1.88511238e-02  | -6.88591301e-02 |
| -1.89459860e-01 | 4.09900129e-01  | -5.97114302e-02 | -1.44015148e-01 |
| 5.42716905e-02  | 4.95139062e-02  | -3.05095583e-01 | -1.13192007e-01 |
| -1.61364481e-01 | -6.29160628e-02 | 3.45439762e-02  | 1.55333444e-01  |
| 1.95764583e-02  | 2.56241411e-02  | -1.43274307e-01 | -8.39604065e-02 |
| -5.21326140e-02 | -1.78045332e-02 | -6.27734745e-03 | -3.98406982e-02 |
| -2.14333445e-01 | -3.46357852e-01 | -1.19998738e-01 | 2.38846600e-01  |
| 5.13705127e-02  | -1.80835202e-01 | -2.56326914e-01 | -1.22217134e-01 |
| 1.00539111e-01  | -2.86362082e-01 | -8.52771923e-02 | 1.08049706e-01  |
| 1.61151603e-01  | 2.05501229e-01  | 3.15578789e-01  | -2.46759713e-01 |
| -5.23773488e-03 | 1.86267287e-01  | 1.42039523e-01  | -7.87023827e-02 |

|                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|
| -2.02376336e-01 | -1.58633310e-02 | 1.34672642e-01  | 2.77339318e-03  |
| 1.82470620e-01  | 2.06106920e-02  | 2.60120839e-01  | -1.97965026e-01 |
| 9.64970961e-02  | -9.34484452e-02 | 6.55125752e-02  | 1.47960521e-02  |
| -7.38896206e-02 | 3.37433405e-02  | -2.33226828e-02 | 2.36117225e-02  |
| 5.19042723e-02  | 3.96772444e-01  | 7.30908439e-02  | -1.75011605e-01 |
| -6.92530125e-02 | -4.54812869e-02 | 1.14286445e-01  | -1.37105547e-02 |
| 1.31576315e-01  | 1.57922506e-01  | -2.04118267e-02 | -3.28725688e-02 |
| -4.20016311e-02 | 8.03523213e-02  | -3.79251316e-02 | 1.37778313e-03  |
| 1.67010620e-01  | -1.45056009e-01 | 2.47669257e-02  | 6.90767122e-03  |
| -8.09295475e-03 | 1.53843850e-01  | -8.51320699e-02 | 1.77630916e-01  |
| 1.18224293e-01  | 1.43131360e-01  | 3.72576177e-01  | 2.79713780e-01  |
| -2.04580557e-02 | 8.12711343e-02  | -1.45523414e-01 | -2.28033975e-01 |
| -1.20879531e-01 | -1.19529903e-01 | -1.72291726e-01 | 1.80338636e-01  |
| -2.42391348e-01 | 1.82836764e-02  | 6.09526150e-02  | 7.96274543e-02  |
| 6.87080622e-02  | -4.80795875e-02 | -1.10261209e-01 | 9.34368744e-02  |
| -6.06875233e-02 | -3.17652710e-02 | 4.84169982e-02  | 7.85343871e-02  |
| 9.90398675e-02  | -9.96591151e-02 | 4.65055481e-02  | 9.12414715e-02  |
| -2.21814975e-01 | -1.14556842e-01 | 9.63571072e-02  | 9.00609642e-02  |
| 2.96440363e-01  | -6.51353225e-02 | -3.06090899e-02 | -2.92082932e-02 |
| 2.36718506e-02  | 5.17096333e-02  | 1.74049437e-01  | -1.00938447e-01 |
| -1.27592012e-01 | 2.13183910e-02  | 2.87751257e-01  | 1.37189953e-02  |
| -1.18777320e-01 | 1.64411202e-01  | -4.44606878e-02 | -1.76178724e-01 |
| -5.17188050e-02 | 4.53070514e-02  | -2.08016098e-01 | 2.23998174e-01  |
| 1.22284181e-01  | -1.59077913e-01 | -9.01502743e-02 | -1.73562884e-01 |
| -2.13511437e-02 | 1.28715813e-01  | -6.35630963e-03 | -3.57450284e-02 |
| 7.89927319e-02  | -2.16908175e-02 | -8.09005722e-02 | 5.77619635e-02  |
| -5.57258166e-02 | 5.71475066e-02  | 4.78018560e-02  | -9.15982574e-02 |
| -1.99919626e-01 | -1.22598335e-01 | -7.48332813e-02 | 7.53318379e-03  |
| 1.05689727e-01  | 2.98688691e-02  | 2.78096199e-01  | -1.38005137e-01 |
| -1.58366084e-01 | 2.61439353e-01  | 2.76638735e-02  | -8.85913968e-02 |
| -4.16143574e-02 | -1.07053369e-01 | -1.45831853e-01 | 3.02815646e-01  |
| 1.19237959e-01  | 2.44743824e-01  | 9.74552855e-02  | 2.73321867e-02  |
| -1.66532338e-01 | 9.36323330e-02  | 7.34953508e-02  | 4.70826514e-02  |
| 2.42118493e-01  | -3.59026819e-01 | -1.88048944e-01 | 1.65457577e-01  |
| -2.11359367e-01 | -2.81181782e-01 | -1.57508045e-01 | 1.38785809e-01  |
| -3.01500648e-01 | -9.64521393e-02 | 2.25491673e-01  | 1.61943603e-02  |
| 1.03646358e-02  | 3.36555503e-02  | -2.88897634e-01 | 1.85248420e-01  |
| -5.56219146e-02 | -1.56955883e-01 | -3.43350500e-01 | 5.51126385e-03  |
| 3.13149090e-03  | 8.45114589e-02  | -1.05394430e-01 | -3.49514395e-01 |
| -2.38996223e-01 | -1.12128435e-02 | 1.47496223e-01  | -1.74420759e-01 |
| -1.23622539e-02 | 1.44026041e-01  | -9.62103158e-03 | -2.03342363e-01 |
| -2.87828565e-01 | -2.70598531e-02 | -7.22167664e-04 | 1.54862285e-01  |
| -4.22951467e-02 | -2.29216143e-01 | 1.10092118e-01  | -5.37884757e-02 |
| 1.56729892e-01  | -3.48104388e-01 | -2.71738738e-01 | 3.32502834e-02  |
| 7.68075809e-02  | 6.56878725e-02  | -1.06658973e-01 | 5.21143840e-04  |
| -5.78524433e-02 | -7.36805424e-03 | -1.85162634e-01 | 2.46318188e-02  |
| 1.23737864e-01  | -1.16908245e-01 | -7.78210536e-02 | 1.19121075e-01  |
| 4.28733639e-02  | -6.87618926e-02 | 1.00417711e-01  | -2.27898762e-01 |
| 5.24059571e-02  | 3.14734615e-02  | -1.67964891e-01 | -5.95557764e-02 |
| -4.54565771e-02 | -1.19392678e-01 | -2.42819294e-01 | 8.78881291e-02  |
| 1.23397559e-01  | -9.52609628e-02 | -1.54749081e-01 | 2.95499898e-03  |
| -3.08234897e-02 | -8.08541849e-02 | 8.85417238e-02  | -1.90988313e-02 |
| 9.88056585e-02  | -4.61402945e-02 | -1.89017177e-01 | -1.91102654e-01 |
| 1.06500797e-01  | -3.97194996e-02 | 2.50703275e-01  | 9.26945433e-02  |
| 6.86516389e-02  | -2.50271767e-01 | 7.18227103e-02  | 8.92523080e-02  |
| -1.07179202e-01 | 1.95978552e-01  | -4.21196595e-02 | -5.93543984e-02 |
| -8.13741609e-02 | -1.39769375e-01 | 6.03454113e-02  | -2.67672297e-02 |

```
-2.45423824e-01 -1.31246774e-02  8.66285432e-03  1.00736395e-02  
-1.69458359e-01  9.87705365e-02 -4.23314497e-02  1.60690367e-01  
 4.53071505e-01  5.98389395e-02 -3.13517034e-01  4.78915777e-03] ->  
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]]
```

In [16]: *# Let's create the models*

```
vocab_size_spooky = len(tokenizer_spooky.index_word.keys())
vocab_size_niet = len(tokenizer_niet.index_word.keys())

neural_lm_spooky = Sequential()
neural_lm_spooky.add(Embedding(vocab_size_spooky, 100, weights=[spooky_embeddings], input_length=4, trainable=False))
neural_lm_spooky.add(Dense(32, activation='tanh'))
neural_lm_spooky.add(Dense(50, activation='relu'))
neural_lm_spooky.add(Dense(len(tokenizer_spooky.index_word.keys()), activation='softmax'))

neural_lm_spooky.compile(loss='categorical_crossentropy', optimizer='sgd')

neural_lm_niet = Sequential()
neural_lm_niet.add(Embedding(vocab_size_niet, 100, weights=[niet_embeddings], input_length=4, trainable=False))
neural_lm_niet.add(Dense(32, activation='tanh'))
neural_lm_niet.add(Dense(50, activation='relu'))
neural_lm_niet.add(Dense(len(tokenizer_niet.index_word.keys()), activation='softmax'))

neural_lm_niet.compile(loss='categorical_crossentropy', optimizer='sgd')
```

```
2020-11-12 20:50:39.202947: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2020-11-12 20:50:39.230558: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x7f9ed2c47e10 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2020-11-12 20:50:39.230577: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
```



```

-----
AttributeError                                Traceback (most recent call last)
<ipython-input-16-3d9093cf0d16> in <module>
      5
      6 neural_lm_spooky = Sequential()
----> 7 neural_lm_spooky.add(Embedding(vocab_size_spooky, 100, weights=
[spooky_embeddings], input_length=4, trainable=False))
      8 neural_lm_spooky.add(Dense(32, activation='tanh'))
      9 neural_lm_spooky.add(Dense(50, activation='relu'))

~/opt/anaconda3/lib/python3.8/site-packages/tensorflow/python/training/
tracking/base.py in _method_wrapper(self, *args, **kwargs)
    455     self._self_setattr_tracking = False # pylint: disable=prot
ected-access
    456     try:
--> 457         result = method(self, *args, **kwargs)
    458     finally:
    459         self._self_setattr_tracking = previous_value # pylint: d
isable=protected-access

~/opt/anaconda3/lib/python3.8/site-packages/tensorflow/python/keras/eng
ine/sequential.py in add(self, layer)
    204         # and create the node connecting the current layer
    205         # to the input layer we just created.
--> 206         layer(x)
    207         set_inputs = True
    208

~/opt/anaconda3/lib/python3.8/site-packages/tensorflow/python/keras/eng
ine/base_layer.py in __call__(self, *args, **kwargs)
    923     # >> model = tf.keras.Model(inputs, outputs)
    924     if _in_functional_construction_mode(self, inputs, args, kwa
args, input_list):
--> 925         return self._functional_construction_call(inputs, args, k
wargs,
    926                                                     input_list)
    927

~/opt/anaconda3/lib/python3.8/site-packages/tensorflow/python/keras/eng
ine/base_layer.py in _functional_construction_call(self, inputs, args,
kwargs, input_list)
    1096         # Build layer if applicable (if the `build` method has
been
    1097         # overridden).
-> 1098         self._maybe_build(inputs)
    1099         cast_inputs = self._maybe_cast_inputs(inputs, input_lis
t)
    1100

~/opt/anaconda3/lib/python3.8/site-packages/tensorflow/python/keras/eng
ine/base_layer.py in _maybe_build(self, inputs)
    2653         # Using `init_scope` since we want variable assignmen
t in
    2654         # `set_weights` to be treated like variable initializ
ation.

```

```

-> 2655         self.set_weights(self._initial_weights)
    2656     else:
    2657         self.set_weights(self._initial_weights)

~/opt/anaconda3/lib/python3.8/site-packages/tensorflow/python/keras/engine/base_layer.py in set_weights(self, weights)
    1821         weight = weights[weight_index]
    1822         ref_shape = param.shape
-> 1823         if not ref_shape.is_compatible_with(weight.shape):
    1824             raise ValueError(
    1825                 'Layer weight shape %s not compatible with provided weight '

```

**AttributeError:** 'Word2VecKeyedVectors' object has no attribute 'shape'

```

In [ ]: # Training the models
        neural_lm_niet.fit(data_generator_niet, epochs=1)
        neural_lm_niet.fit(data_generator_niet, epochs=1)

        print(neural_lm_spooky.summary())
        print(neural_lm_niet.summary())

```

## e) Generate Sentences

```

In [ ]: # generate a sequence from the model
def generate_seq(model, tokenizer, seed, n_words, embedding):
    '''
    Parameters:
        model: your neural network
        tokenizer: the keras preprocessing tokenizer
        seed: [w1, w2, w(n-1)]
        n_words: generate a sentence of length n_words
    Returns: string sentence
    '''
    output = seed
    while len(output) < n_words:
        embedded = np.concatenate([embedding.get_vector(word) for word i
n output[-NGRAM:]])
        prob_distr = model.predict(embedded)
        next_word_idx = np.argmax(prob_distr)
        output.append(tokenizer.word_index[next_word_idx])

    return output

niet_seed = [
    ["in", "the", "belief"],
    ["we", "need", "it"],
    ["tried", "to", "describe"],
    ["for", "life", "and"],
    ["tell", "me", "that"],
    ["great", "historical", "movement"],
    ["at", "all", "costs"],
    ["i", "gain", "an"],
    ["these", "thoughts", "are"],
    ["as", "a", "fault"],
    ["even", "if", "it"],
    ["be", "a", "virtue"],
    ["a", "child", "of"],
    ["i", "must", "admit"],
    ["man", "cannot", "see"],
    ["simply", "to", "live"],
    ["to", "live", "without"],
    ["the", "beast", "wants"],
    ["but", "he", "forgets"],
    ["i", "always", "forget"]]

spooky_seed = [
    ["if", "a", "fire"],
    ["as", "i", "urged"],
    ["the", "tone", "metaphysical"],
    ["the", "offspring", "of"],
    ["persuading", "the", "widow"],
    ["when", "i", "arose"],
    ["tears", "in", "my"],
    ["nothing", "was", "elicited"],
    ["the", "lost", "inheritance"],
    ["could", "almost", "see"],
    ["freak", "of", "fancy"],
    ["grapple", "not", "now"],
    ["upon", "the", "figure"],
    ["restrain", "your", "impatience"],

```

```

        ["he", "is", "dead"],
        ["i", "felt", "much"],
        ["and", "this", "fall"],
        ["during", "this", "shocking"],
        ["was", "he", "murdered"],
        ["darkness", "had", "no"]]

print("Spooky sentences:")
print("=====")
for each_seed in spooky_seed:
    print(" ".join(generate_seq(neural_lm_spooky, tokenizer_spooky, each_
_seed, 20, spooky_embeddings)))

print("Philosophical sentences:")
print("=====")
for each_seed in niet_seed:
    print(" ".join(generate_seq(neural_lm_niet, tokenizer_nit, each_seed
, 20, niet_embeddings)))

```

## **Shannon's method bigram generated sentences**

**Spooky:**

<s> in <UNK> was <UNK> said was <UNK> bore the thickly <UNK> <UNK> <UNK> of all <UNK> </s>

<s> and his <UNK> of this two <UNK> brought to mind was born even to <UNK> u p to <UNK> of my rage and when i <UNK> close to <UNK> i have ever presented itself and of the <UNK> <UNK> <UNK> circumstance from <UNK> <UNK> from the <UNK> <UNK> <UNK> so that it entirely the <UNK> the <UNK> of all the <UNK> </s>

<s> the more and threw himself on a <UNK> into <UNK> <UNK> <UNK> <UNK> <UNK> in all through the latter <UNK> <UNK> <UNK> we <UNK> <UNK> of <UNK> at some <UNK> and <UNK> <UNK> <UNK> of this matter of <UNK> and love in this she di d he <UNK> the very heart of some <UNK> and after dark and knew i had </s>

<s> but during the <UNK> de la <UNK> i <UNK> of <UNK> these the <UNK> appear ance of genius is composed </s>

<s> the <UNK> with the house and <UNK> that it <UNK> at some people have rea lly had <UNK> one life might have of us only the sky that death was <UNK> an d <UNK> </s>

<s> <UNK> and that body or <UNK> </s>

<s> when they also a <UNK> and the <UNK> and <UNK> within </s>

<s> all but by any <UNK> its very <UNK> shall <UNK> the word by which could <UNK> less <UNK> <UNK> of the <UNK> these <UNK> <UNK> <UNK> out for in the former night fell upon the <UNK> <UNK> <UNK> <UNK> stream or if this time w hile his <UNK> <UNK> idris <UNK> the <UNK> </s>

<s> <UNK> <UNK> and <UNK> </s>

<s> in the <UNK> <UNK> stream </s>

<s> <UNK> in with <UNK> he <UNK> <UNK> the <UNK> <UNK> during this as it </s>

>

<s> if this as with them to advance with such <UNK> of way the <UNK> of <UNK> > </s>

<s> thus <UNK> any <UNK> <UNK> there <UNK> have none of the fact that an ord inary <UNK> with <UNK> i am <UNK> have of the <UNK> <UNK> <UNK> at <UNK> was a <UNK> <UNK> <UNK> <UNK> of may the <UNK> close to a <UNK> of all night </s>

>

<s> at all <UNK> to <UNK> <UNK> of mankind ever done for he seemed to <UNK> many days of pleasure </s>

<s> i thought nothing was <UNK> you looking <UNK> it entirely <UNK> and when their <UNK> many days <UNK> and i <UNK> </s>

<s> upon <UNK> into some time he gave a body or <UNK> and <UNK> <UNK> friend i have your <UNK> i <UNK> <UNK> to the fact that <UNK> and <UNK> more <UNK> that <UNK> her </s>

<s> <UNK> that ever <UNK> <UNK> <UNK> </s>

<s> i therefore <UNK> <UNK> </s>

<s> i again to see shadowy <UNK> on the <UNK> <UNK> in <UNK> at their <UNK> <UNK> him neither <UNK> <UNK> <UNK> up to the <UNK> over all <UNK> than adv ance <UNK> my heart and <UNK> in my next night when they shall know <UNK> th e <UNK> <UNK> <UNK> in the more <UNK> of <UNK> therefore <UNK> <UNK> <UNK> </s>

<s> the black <UNK> of dreams but monsieur <UNK> <UNK> before <UNK> object a lmost of the black <UNK> of what you cried read it stood <UNK> </s>

<s> if the <UNK> he seemed to <UNK> <UNK> of our first <UNK> in <UNK> was mo

re of my friend will <UNK> we placed in that it or <UNK> for a <UNK> of a <UNK> perhaps it not <UNK> soon after a day he <UNK> as with them one <UNK> </s>

<s> this <UNK> that one thing <UNK> her and <UNK> <UNK> for come </s>

<s> the most vividly desire to which by fellows to form a feeling the whole strength of <UNK> or gave it all be with <UNK> was hope one of <UNK> and <UNK> with more <UNK> </s>

<s> should <UNK> </s>

<s> the latter and in this as a <UNK> <UNK> in that while the shadowy <UNK> <UNK> of the <UNK> gave me from the <UNK> <UNK> were <UNK> shall be later period the fact that <UNK> spirit when the <UNK> and some day shall <UNK> to <UNK> and <UNK> over the <UNK> and love of fancy in <UNK> <UNK> </s>

<s> the tide is <UNK> the whole country as the <UNK> as the <UNK> and <UNK> <UNK> was <UNK> <UNK> it or a <UNK> over the <UNK> </s>

<s> <UNK> gave the <UNK> not <UNK> with a last <UNK> </s>

<s> his fellows to <UNK> of way </s>

<s> did <UNK> <UNK> if this shocking period stood in this second <UNK> <UNK> i made shocking period stood like <UNK> </s>

<s> was thickly <UNK> of the <UNK> i have your <UNK> a <UNK> </s>

<s> all <UNK> of the <UNK> <UNK> <UNK> of <UNK> to such <UNK> </s>

<s> yet he had been in <UNK> <UNK> within the <UNK> of these we should <UNK> <UNK> his nod i am not <UNK> of the manuscript all </s>

<s> he gave themselves the <UNK> </s>

<s> the respect and <UNK> of his <UNK> <UNK> with <UNK> of <UNK> of <UNK> i am <UNK> <UNK> of that which <UNK> <UNK> of dreams and of her spirit </s>

<s> upon my eyes </s>

<s> <UNK> that are adapted she indeed his <UNK> <UNK> of his <UNK> <UNK> in the <UNK> and part of the <UNK> </s>

<s> the <UNK> <UNK> </s>

<s> <UNK> into the <UNK> that <UNK> without them had by the evening <UNK> of the <UNK> </s>

<s> before the <UNK> since the free </s>

<s> <UNK> like <UNK> </s>

<s> these we had broken down the plague itself and <UNK> became the <UNK> it would become food for the object should be later <UNK> <UNK> who shall <UNK> of the <UNK> idea can do nothing was the <UNK> of them a <UNK> and <UNK> <UNK> and love in the word to <UNK> time while those of other eyes </s>

<s> the <UNK> they now the <UNK> was that <UNK> <UNK> over and even more than those <UNK> that he <UNK> <UNK> <UNK> <UNK> of her <UNK> on the car itself </s>

<s> and <UNK> of a <UNK> in its <UNK> way the <UNK> of every <UNK> and when the part from the <UNK> of his grief his grief his <UNK> </s>

<s> and <UNK> since <UNK> we shall <UNK> and horror <UNK> that i did at least dark is <UNK> <UNK> of a <UNK> he <UNK> <UNK> gave me for what <UNK> the <UNK> <UNK> <UNK> the <UNK> placed <UNK> of <UNK> page <UNK> of our <UNK> separate and <UNK> be received <UNK> </s>

<s> monsieur <UNK> to see you let me and then of <UNK> </s>

<s> it should <UNK> <UNK> sighing as in the <UNK> out of <UNK> <UNK> <UNK> and that body but although i have ever <UNK> in the throne of the <UNK> witho

ut them one or more <UNK> i <UNK> was the next <UNK> is at the <UNK> <UNK> <UNK> <UNK> her spirit </s>

<s> when any <UNK> and <UNK> my <UNK> or you looking even more of <UNK> <UNK> <UNK> <UNK> <UNK> his <UNK> as <UNK> had seen the <UNK> the society a <UNK> forms that my <UNK> when i am <UNK> less <UNK> yet <UNK> and <UNK> <UNK> has it on the very cause of that my father thought it entirely <UNK> by inheritance of <UNK> <UNK> a few <UNK> the earth again to each of the day at least <UNK> </s>

<s> you must have of the balloon the <UNK> me view with her own mind the <UNK> <UNK> le <UNK> by fellows to <UNK> of the <UNK> forms an exceedingly <UNK> <UNK> the <UNK> less <UNK> of <UNK> as it on the inheritance the name as far as <UNK> <UNK> <UNK> of <UNK> to <UNK> <UNK> to town had some <UNK> during all objects <UNK> his <UNK> very <UNK> above that my <UNK> in the <UNK> <UNK> proved themselves the <UNK> to each of the <UNK> with them a <UNK> along the room </s>

<s> i sought <UNK> <UNK> and <UNK> or <UNK> by dreams and horror of <UNK> </s>

<s> the <UNK> <UNK> de <UNK> our first of my <UNK> <UNK> my peculiar <UNK> of the least <UNK> would become her and thus <UNK> <UNK> such <UNK> it could not to which i sing to <UNK> <UNK> and even <UNK> not how much of the <UNK> </s>

## Nietzsche:



<s> forces the most forbidden truths the form of my thuringian virgin </s>  
<s> man of letters of virtue </s>  
<s> inverted </s>  
<s> scarcely feel very far </s>  
<s> oh this effect may be an impressive and even excited awe of pity and the  
y discover the justice is then the philosopher and tasteful in our inner <UN  
K> itself a very difficult to become weary and oppressed down-trodden humani  
ty and the </s>  
<s> all who treat indulgently i might almost with the </s>  
<s> it is a soft benevolent and we should not yet they both of every woman u  
nto themselves when man <UNK> element the weak good-natured smile </s>  
<s> and even as the world </s>  
<s> in this motleyness </s>  
<s> and goodness sake and a question dost thou my way </s>  
<s> downwards </s>  
<s> if self deception great </s>  
<s> it is verily did not in my six-and-thirtieth year </s>  
<s> civilisation any rate certain paragraphs </s>  
<s> he </s>  
<s> they had to give and error in epochs nations and the </s>  
<s> whatever objects he spake the </s>  
<s> german culture it comes to be made is useless burden and the skeptic con  
sole him obey now <UNK> it and the proposition </s>  
<s> immediately and still be not furious to my brother said then must accoun  
t </s>  
<s> strengthened themselves in german soldiers </s>  
<s> there came afterwards being lost every one must be </s>  
<s> to which the morality in revolution he would presuppose </s>  
<s> sense some order to require a normal phenomena </s>  
<s> idea of fine ears what deceives himself decides in injuring the beginnin  
g is to do everything evil beyond as le plus ultra of the </s>  
<s> on the average even more </s>  
<s> one example being is a method of ourselves once be only in </s>  
<s> and in </s>  
<s> plainly distinguished by the excessive brooding natures operate which is  
the </s>  
<s> at me i look down to new beauty of children with innate anticipation not  
find ourselves on the discord and eccentric concept sin </s>  
<s> where it alienates have hope and the commedia <UNK> such a profounder na  
tures whoever <UNK> </s>  
<s> to </s>  
<s> friend and </s>  
<s> even time it was his neighbour so great majority that is the universal a  
nd systematising </s>  
<s> merged into a love and their misery of the extirpation </s>  
<s> of action of thirteen the religio-moral </s>  
<s> the fact that commerce have the </s>

## f) Compare your generated sentences

In [ ]:

## Sources Cited

1. <https://github.com/DmitryUlyanov/Multicore-TSNE> (<https://github.com/DmitryUlyanov/Multicore-TSNE>) (FAST T-SNE)
2. Kulshethra, Ria. "NLP 101: Word2Vec — Skip-gram and CBOW". Accessed November 8, 2020. <https://towardsdatascience.com/nlp-101-word2vec-skip-gram-and-cbow-93512ee24314> (<https://towardsdatascience.com/nlp-101-word2vec-skip-gram-and-cbow-93512ee24314>) (Used to answer the questions about CBOW and skip-gram.)
3. Wang, Bin et al. "Evaluating Word Embedding Models: Methods and Experimental Results". <https://arxiv.org/abs/1901.09785> (<https://arxiv.org/abs/1901.09785>)
4. Winastwan, Rubin. "Visualizing Word Embedding with PCA and t-SNE". Accessed November 8, 2020. <https://towardsdatascience.com/visualizing-word-embedding-with-pca-and-t-sne-961a692509f5> (<https://towardsdatascience.com/visualizing-word-embedding-with-pca-and-t-sne-961a692509f5>)
5. Kurita, Keita. "Paper Dissected: “Visualizing Data using t-SNE” Explained". Accessed November 8, 2020. <https://mlexplained.com/2018/09/14/paper-dissected-visualizing-data-using-t-sne-explained/> (<https://mlexplained.com/2018/09/14/paper-dissected-visualizing-data-using-t-sne-explained/>)
6. Brownlee, Jason. "How to Develop a Word-Level Neural Language Model and Use it to Generate Text". Accessed November 8, 2020. <https://machinelearningmastery.com/how-to-develop-a-word-level-neural-language-model-in-keras/> (<https://machinelearningmastery.com/how-to-develop-a-word-level-neural-language-model-in-keras/>)

In [ ]: